ICCBR 2017

Trondheim, Norway 26-28 June 2017

The Twenty-Fifth International Conference on Case-Based Reasoning (ICCBR 2017)

Workshop Proceedings

Antonio A. Sanchez Ruiz-Granados and Anders Kofod-Petersen
(Editors)

Preface

We are pleased to present the Workshop Proceedings of the Twenty-fifth International Conference on Case-based Reasoning (ICCBR-17), held on June 26$^{\text{th}}$ - 29$^{\text{th}}$ in Trondheim, Norway.

This years brings three workshops: the Computational Analogy Workshop (CAW), the Case-based Reasoning and Deep Learning Workshop (CBRDL), and the workshop on Process-oriented Case-based Reasoning (PO-CBR). This years also features the proud tradition of the Doctoral Consortium and the Computer Cooking Contest.

We would like to thank all who contributed to the success of this workshop program, especially the authors and presenters who provided the research insights comprising the essential substance of the workshops. We would also like to thank all of the program committee members for their contributions leading to high-quality submissions, and especially lthe workshop organisers for their hard work over the past year developing a compelling collection of workshop programs.

Our special thanks go to the program chairs, David W. Aha and Jean Lieber; the local chair, Odd Erik Gundersen; and the publicity chair, Kerstin Bach, for their continuous support, efforts in planning the event, and assistance with producing the proceedings.

We hope that authors and other participants enjoy and are invigorated by this year's workshop program and also find valuable resources in these proceedings for furthering their research. We look forward to a fruitful exchange of ideas in Trondheim.

June 2017
Trondheim

Antonio A. Sanchez
Ruiz-Granados, Anders
Kofod-Petersen
(Workshop Chairs)

Table of Contents

3rd Workshop on Process-oriented Case-based Reasoning: PO-CBR@ICCBR-2017

25th International Conference on Case-Based Reasoning - Doctoral Consortium 2017

Computer Cooking Contest 2017

Preface

Computational Analogy and Case-Based Reasoning (CBR) are closely related research areas. Both employ prior cases to reason in complex situations with incomplete information. Analogy research often focuses on modeling human cognitive processes, the structural alignment between a case/source and target, and adaptation/abstraction of the analogical source content. While CBR research also deals with alignment and adaptation, the field tends to focus more on retrieval, case-base maintenance, and pragmatic solutions to real-world problems. However, despite their obvious overlap in research goals and approaches, cross communication and collaboration between these areas has been progressively diminishing. CBR and Analogy researchers stand to benefit greatly from increased exposure to each others work and greater cross-pollination of ideas. The objective of this workshop is to promote such communication by bringing together researchers from the two areas, to foster new collaborative endeavors, to stimulate new ideas and avoid reinventing old ones.

In this second edition of the ICCBR Workshop on Computational Analogy, 8 papers have been selected for presentation (on 9 submissions) from researchers coming from USA, France, and Japan).

A first set of contributions concern formal aspects of the analogical inference, its complexity and evaluation. Henri Prade and Gilles Richard provide a state of the art on analogical-proportion based inference. As recent results show that affine Boolean functions can be predicted without error by means of analogical proportions, the authors discuss how one might take advantage of this result to refine the scope of application of the analogical-proportion based inference to subparts of a Boolean function that may be assumed to be “locally” linear.

Also contributing to formal analogy research, Yves Lepage addresses the problem of answering analogy questions of the type $A:B::C:D$ between word forms where the unknown is D. The author proposes to add a new criterion for partial determination of the solution to an analogy question: the pairwise indices of the positions of the characters. A character-position matrix is built which assigns a probability to each character and position in the answer D of an analogy question $A:B::C:D$.

Two papers adress the issue of the evaluation of an analogical inference. The work of Pierre-Alexandre Murena and his colleagues aim at testing the hypothesis that the relevance of an analogical solution can be measured by the complexity of the analogy. In order to compute the complexity, they propose some specifications for a prototype language used to describe analogies in a basic alphanumeric micro-world domain. Joseph Blass and his colleagues take another approach to evaluation. Starting from the observation that inferred facts are, even if the reasoning technique is sound, only as accurate as the assumptions upon which they are based, they propose a domain-independent method to evaluate inferences for analogical reasoning, via a prototype system.

Another set of contributions concern analogy for reuse and adaptation. Scott Friedman and his colleagues explore how analogical mappings can be used to help humans and computers negotiate to define shared goals and collaborate over the fulfillment of those goals. Their application domain is plan localization, the problem of establishing the set of steps within the plan that are candidates (potentially after some adaptive repair actions) for next actions given the worlds unforeseeen changes. Fadi Badra develops a qualitative modeling approach of the case-based analogical inference, and proposes a language to represent and symbolically reason upon differences between cases. This language can be used to represent both similarity paths and adaptation rules.

A last contribution uses machine learning techniques to learn how to solve analogical equations in the domain of Natural Language Processing. Rashel Fam and Yves Lepage analyse the characteristics of a data structure called an "analogical grid", which can be used to predict new word forms (morphological forms) from purely surface level observations of the words found in text. In particular, the saturation of analogical grids is measured against their size. Reported results show that the logarithm of the saturation of an analogical grid is linear in the logarithm of its size, and the relation between the saturation and the size of an analogical grid is almost independent of the size, the genre and the language of a text.

June 2017

Fadi Badra and Tarek Besold
Program Chairs
Computational Analogy Workshop 2017

Organization

Organizing Commitee

Fadi Badra (Université Paris 13, France)
Tarek Besold (University of Bremen, Germany)
Joe Blass (Northwestern University, USA)
Tesca Fitzgerald (Georgia Institute of Technology, USA)
Vincent Letard (LIMSI CNRS, France)

Program Commitee

Myriam Bounhas (LARODEC, Tunisia)
Mark Burstein (SIFT, USA)
Hernan Casakin (Ariel University, Israel)
Amaresh Chakrabarti (Indian Institute of Science, India)
Antoine Cornuéjols (LRI, Orsay, France)
Mark Finlayson (FIU, USA)
Kenneth Forbus (Northwestern University, Chicago, USA)
Scott Friedman (SIFT, USA)
Katherine Fu (Georgia Institute of Technology, Atlanta, USA)
Bipin Indurkhya (Jagiellonian University, Poland)
Mark Keane (UC Dublin, Ireland)
Philippe Langlais (Université de Montreal, Canada)
Yves Lepage (Waseda University, Japan)
Abhijit Mahabal (Google, USA)
Santiago Ontanon (Drexel University, USA)
Henri Prade (IRIT, Toulouse, France)
Gilles Richard (Universit Paul Sabatier, Toulouse, France)
Emmanuel Sander (Universit Paris 8, Paris, France)
Ute Schmid (Universität Bamberg, Germany)
Steven Schockaert (Cardiff University, UK)
Christian Schunn (University of Pittsburg, USA)
Bryan Wiltgen (Georgia Institute of Technology, USA)

A study of the saturation of analogical grids agnostically extracted from texts

Rashel Fam and Yves Lepage *

IPS, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, 808-0135 Fukuoka-ken, Japan
fam.rashel@fuji.waseda.jp, yves.lepage@waseda.jp

Abstract. Analogical grids aim to capture the organization of the lexicon of a language. We conduct experiments on analogical grids extracted in four different languages with different morphological richness. We study the saturation of analogical grids against their size. We observe that the logarithm of the saturation of an analogical grid is linear in the logarithm of its size. More surprisingly, the coefficients of this log-log linear relation are extremely close across all four languages, even when the size or the genre of the corpus vary.

Keywords: analogical grids, saturation, organization of lexicon.

1 Introduction and background

show	:	*shows*	:	*showing*	:	*showed*
walk	:	*walks*	:	*walking*	:	*walked*
open	:	*opens*	:	*opening*	:	
study	:		:	*studying*	:	
read	:	*reads*	:	*reading*	:	

makan	:	*dimakan*	:	*memakan*	:	*makanan*
minum	:	*diminum*	:	*meminum*	:	*minuman*
main	:		:		:	*mainan*
beli	:	*dibeli*	:		:	

Fig. 1. Analogical grids in English (*left*) and Indonesian (*right*).

Figure 1 shows two examples of analogical grids, one in English, the other one in Indonesian. Such analogical grids may be automatically constructed from the set of words contained in a text. Each cell in an analogical grid either contains a word form or is empty. As exemplified in Figure 1 (*left*), a column (or a row) in an analogical grid usually exhibits similar word forms for different words: e.g., infinitive, present 3rd person singular, present participle, etc. for different English verbs on the *left* of Figure 1. Analogical grids are not paradigm tables,

* This work was supported by a JSPS Grant, Number 15K00317 (Kakenhi C), entitled Language productivity: efficient extraction of productive analogical clusters and their evaluation using statistical machine translation.

i.e., they are not the result of a linguistic formalization with explicit lexemes and exponents as in standard works in morphology, but they constitute a preliminary step in that direction. Analogical grids too give a compact view of the organization of the lexicon, but they are the output of an empirical procedure, e.g., the one introduced in [4].

Analogical grids can be used to study word productivity in a given language as in [12, 9, 6]. They can also be used to make comparisons across languages as in [4], where the goal is to explain unseen words by using analogical grids automatically built from the set of all words contained in texts in 12 different languages.

In this paper, we report an interesting phenomenon observed when building analogical grids in various different languages using the method in [4]. This phenomenon relates the saturation of the obtained analogical grids to their size. The experimental results show that the coefficients which characterize the relation would not be influenced by the size, the genre or the language of the texts used.

The paper is organized as follows: Section 2 introduces basic notions related to analogical grids. Section 3 presents our experiments on four languages with different richness in morphology. It analyzes the results and explores the relationship between the saturation and the size of analogical grids. Section 4 presents further experiments to inquire the relation. Section 5 gives conclusion.

2 Basic notions

In this section, we mathematically define the basic notions related to analogical grids. The method to extract such analogical grids has already been presented elsewhere [8, 4].

2.1 Illustration with toy data

Anto memakan nasi dan meminum air. Nasi itu dibeli di pasar. Di pasar, Anto melihat mainan. Anto senang main bola. Setelah main, Anto suka minum es dan makan cilok. Makanan dan minuman itu juga dia beli di pasar. Es dan cilok memang enak dimakan dan diminum selesai olahraga.

air anto ***beli*** *bola cilok dan di dia* ***dibeli dimakan diminum*** *enak es itu juga* ***main mainan makan makanan*** *melihat* ***memakan*** *memang* ***meminum minum minuman*** *nasi olahraga pasar selesai senang setelah suka*

Fig. 2. A text in Indonesian (*above*) and the list of words extracted from it (*below*). Words appearing in Figure 1 (*right*) are boldfaced.

The top of Figure 2 is a forged example text in Indonesian, a language which is known for its relative richness in derivational morphology. We intentionally do not give its translation into English to place the reader in the agnostic position of the computer in front of such data. The list of words, sorted in lexicographic order, that can be extracted from this text, is given at the bottom of Figure 2.

From this word list, some commonalities between words can be identified at a glance. An example is the word *makan* and the word *makanan*. Another is the words *bola* and *beli* which share the same consonants in the same order: *b* and *l*. However, the existence of only one pair is not enough to support the evidence that two words are actually in relation one with the other. On the contrary, for the words *makan* and the word *makanan*, the same *ratio* is seen to hold between several other word pairs from the same text, like *minum* and *minuman*, or *main* and *mainan*. These actually reflect a phenomenon in Indonesian morphology by using the suffix *-an* which builds a noun from active verb.

In standard linguistics, a systematization of these relationships between word forms is given by paradigm tables, which is the result of linguistic formalisation. Here, we agnostically extract analogical grids relying on a formal relationship between words, proportional analogy. The right part of Figure 1 shows the analogical grid extracted from the set of words given in Figure 2.

2.2 Analogical grids

An analogical grid is a table of dimension $M \times N$ as defined by Formula (1). As illustrated by Figure 1, analogical grids extracted from texts usually contain empty cells. (Caution: there is no importance in the order of lines or rows.)

$$\begin{array}{l} P_1^1 : P_1^2 : \cdots : P_1^m \\ P_2^1 : P_2^2 : \cdots : P_2^m \\ \;\vdots \quad \vdots \qquad\quad \vdots \\ P_n^1 : P_n^2 : \cdots : P_n^m \end{array} \quad \stackrel{\Delta}{\Longleftrightarrow} \quad \begin{array}{r} \forall (i,k) \in \{1, \ldots, n\}^2, \\ \forall (j,l) \in \{1, \ldots, m\}^2, \\ P_i^j : P_i^l :: P_k^j : P_k^l \end{array} \tag{1}$$

The definition of analogical grids in Formula (1) implies that for any four word forms at the intersection of two rows and two columns form a proportional analogy between sequences of characters [7,13]. A proportional analogy is defined as a relationship between four objects where two properties are met:

(a) equality of ratios (defined hereafter) between the first and the second terms on one hand, and the third and the fourth terms on the other hand, and
(b) exchange of the means (the second and the third terms can always be exchanged).

$$A : B :: C : D \quad \stackrel{\Delta}{\Longleftrightarrow} \begin{cases} A : B = C : D \\ A : C = B : D \end{cases} \tag{2}$$

According to Formula (1), we can get many analogies from analogical grids in Figure 1. Figure 3 shows three of them.

We define the ratio between two words in Formula (3) as a vector of features made up of all the differences in number of occurrences in the two words, for all

$$\textit{makan} : \textit{makanan} :: \textit{main} : \textit{mainan}$$
$$\textit{makan} : \textit{memakan} :: \textit{minum} : \textit{meminum}$$
$$\textit{minum} : \textit{diminum} :: \textit{beli} : \textit{dibeli}$$

Fig. 3. Some analogies extracted from analogical grid in Figure 1 (*right*).

the characters, whatever the writing system, plus, the distance between the two words.

$$A : B \triangleq \begin{pmatrix} |A|_a - |B|_a \\ |A|_b - |B|_b \\ \vdots \\ |A|_z - |B|_z \\ d(A,B) \end{pmatrix} \qquad \textit{makan} : \textit{makanan} \triangleq \begin{pmatrix} -1 \\ 0 \\ \vdots \\ 0 \\ 2 \end{pmatrix} \tag{3}$$

In Formula (3), the notation $|S|_c$ stands for the number of occurrences of character c in string S. The last dimension, written as $d(A,B)$, is the edit distance between the two strings. This indirectly gives the number of common characters appearing in the same order in A and B.[1]

The above definition of ratios captures prefixing and suffixing. Although we do not show it here, this definition also captures parallel infixing or interdigitation, well-known phenomena in semitic languages [1, 14]. However, reduplication or repetition (e.g. consonant spreading) are not captured by this definition.

$$\overset{\textit{makan} : \textit{makanan}}{\begin{pmatrix} -1 \\ 0 \\ \vdots \\ 0 \\ 2 \end{pmatrix}} = \overset{\textit{main} : \textit{mainan}}{\begin{pmatrix} -1 \\ 0 \\ \vdots \\ 0 \\ 2 \end{pmatrix}} \quad \& \quad \overset{\textit{makan} : \textit{main}}{\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 3 \end{pmatrix}} = \overset{\textit{makanan} : \textit{mainan}}{\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 3 \end{pmatrix}}$$
$$\Rightarrow$$
$$\textit{makan} : \textit{makanan} :: \textit{main} : \textit{mainan}$$

Fig. 4. The two ratios between pairs of words for the first analogy in Figure 3.

This formal definition of word ratio in Formula (3) gives the same vector for the ratios *makan* : *makanan* , *makan* : *namakan* , and *makan* : *mnaakan* . This is due to the use of insertion and deletion as the only edit operations.

The purpose of working with analogical grid, and not only with individual analogies, is that Formula (1) imposes more constraints for a word form to enter

[1] The only two edit operations used are insertion and deletion, hence, $d(A,B) = |A| + |B| - 2 \times s(A,B)$. $|S|$ denotes the length of a string S and $s(A,B)$ is the length of the longest common sub-sequence (LCS) between A and B.

a grid: a word form in a grid must satisfy all analogy relationship with all surrounding word forms in the grid. The word form *makanan* in the analogical grid of Figure 1 (right) is the only word form which fits in, among *makanan*, *namakan*, or *mnaakan*. For example, as proved below, using the words *main* and *mainan* from the analogical grid, the inequality between the ratios $makan : main$ and $namakan : mainan$ implies that there is no analogy between these four words. The same holds for the word form *mnaakan*. In all these cases, the inequality comes from different edit distance values.

$$\underset{}{\overset{makan : main}{\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 3 \end{pmatrix}}} \neq \overset{namakan : mainan}{\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 5 \end{pmatrix}} \quad \Rightarrow makan : main :\not{:} \; namakan : mainan$$

The above discussion shows that there should be a relationship between the size of the analogical grids and the freedom in filling an empty cell in an analogical grid.

2.3 Size and saturation of analogical grids

We simply define the size of an analogical grid as its number of rows multiplied by its number of columns. The analogical grids in Figure 1 has a size of $4 \times 5 = 20$ (*left*) and $4 \times 4 = 16$ (*right*) respectively.

Let us now turn to the number of empty cells of an analogical grid, or rather the number of non-empty cells which we call its *saturation*[2]. We compute it using Formula (4) which will give a saturation of 80 % (*left*) and 75 % (*right*) for Figure 1.

$$\text{Saturation} = 100 - \frac{\text{Number of empty cells } \times 100}{\text{Total number of cells}} \tag{4}$$

3 Experiments

3.1 Data used

We carried out experiments on a multilingual parallel corpus created from the translation of the Bible collected by Christodoulopoulos[3] [10]. We selected four languages with different richness in morphology: English, Russian, Modern Greek, and Indonesian. The reason for using a multilingual parallel corpus is the need to draw conclusions across different languages in a reliable way. Table 1 presents statistics on the corpus. For each text in each language, we first extracted the list of all words, and finally built all analogical grids.

[2] In [2, p. 79], saturation is the maximal proportion of word forms attested for any one lemma of a given paradigm. Here we use the term for each entire grid.

[3] http://homepages.inf.ed.ac.uk/s0787820/bible/

Language	# tokens (N)	# types (V)	Length of types avg±std. dev.	# grids	Time (h:min)
English	792,074	12,498	7.03 ± 2.18	12,855	45
Indonesian	648,606	15,641	7.84 ± 2.63	25,752	2:04
Modern Greek	706,771	36,786	8.49 ± 2.49	69,173	11:03
Russian	560,524	47,226	8.26 ± 2.73	60,035	10:34

Table 1. Statistics on the Bible corpus and number of analogical clusters and number of analogical grids produced in each language with the time needed to produce them

3.2 Analogical grids obtained

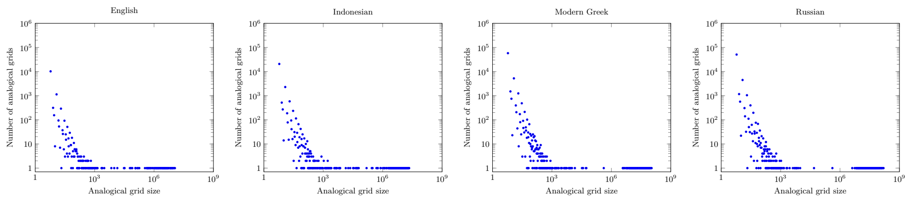


Fig. 5. Number of analogical grids with the same size in each language. Logarithmic scale on both axes. From *left* to *right:* English, Indonesian, Modern Greek and Russian. Same ranges along the axes for all languages.

Table 1 shows the number of analogical grids produced in each language. These numbers show that English produced the lowest number of analogical grids. Indonesian produced twice as many tables as English. Modern Greek and Russian produced five times more tables than English. Modern Greek produced a larger amount of analogical grids than Russian despite its lesser number of analogical clusters. To summarize, languages with poorer morphology tend to produce less analogical grids than languages with richer morphology, which meets intuition.

Let us recall that, by construction, on the contrary to many previous works in morphological induction [11, 5, 3, etc.], our analogical grids do not contain in any way information about word frequency, word context, nor the frequency or distribution of morphemes or the like.

3.3 Size and saturation of analogical grids

The graphs at the bottom of Figure 5 show the number of analogical grids with the same sizes in each language. Most of the analogical grids have a small size. The number of analogical grids with the same size decreases gradually as the size increases. Languages with a richer morphology produce bigger analogical grids in average and also more analogical grids for a given size. All of this meets intuition.

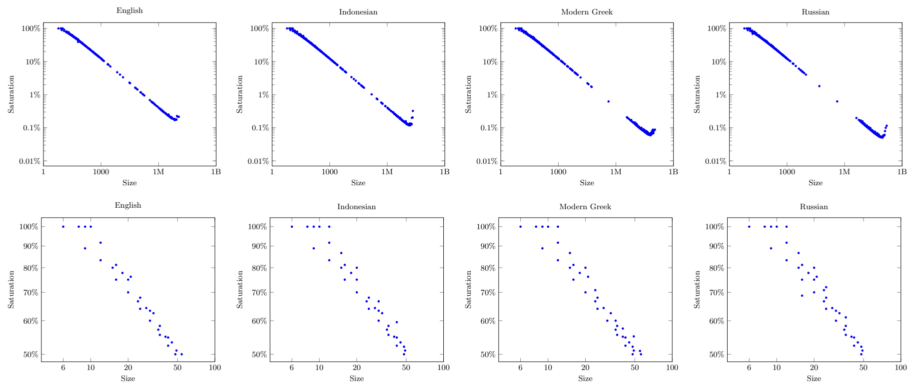


Fig. 6. Saturation of analogical grids against size in each language. From *left* to *right:* English, Indonesian, Modern Greek and Russian. Algorithmic scale on both horizontal (size) and vertical (saturation) axes. Saturation (in ordinates) in the range $[0\,\%, 100\,\%]$ *(top)* and in the range $[50\,\%, 100\,\%]$ *(bottom)*. Same ranges along the horizontal axes for all languages for the same range of saturation.

We now turn to the study of the saturation of analogical grids compared to their size. The top of Figure 6 shows saturation against size for analogical grids in each language. Analogical grids with smaller sizes tend to have higher saturation. Some tables are extremely sparse. Because of the logarithmic scale on the y-axis, the bottom half is for tables with a saturation less than 1 %.

In all cases, the plots exhibit a similar linear shape in logarithmic scale across all languages. This would correspond to Formula (5). We confirmed the similarity by the computation of the coefficients a and b for each language, as obtained by the least squares method. These coefficients are presented in Table 2. They are almost the same in all languages.

$$\log(\text{saturation}) = a \times \log(\text{size}) + b \tag{5}$$

As mentioned in Section 2.2, intuitively, analogical grids with higher saturation are more reliable to fill in because there are more word forms around the empty cells as supporting evidence. However, it may not always be the case. For instance, an analogical grid for regular English verbs extracted from any text is very hollow but empty cells can be filled in a reliable way.

4 Discussion and further experiments

Let us make a first remark on the type of the observed relation. This is not yet another instance of a Zipfian law, because, in the present case, the objects are not ranked individually according to their frequency (number of occurrences). In a Zipfian law, the x-axis stands for the list of individual objects ranked by frequency. Recall also that our analogical grids do not encapsulate any information about the frequency of individual words whatsoever. In our graphs, two analogical grids with the same size have the same abscissa. If they also have the same saturation, they have the same ordinate and are thus plotted as the same point.

Language	Data and size	Range for saturation			
		[0%,100 %]		[50%,100 %]	
		a	b	a	b
English	Bible 100.0 %	-0.480	0.510	-0.366	0.332
	50.0 %	-0.479	0.507	-0.372	0.343
	25.0 %	-0.476	0.499	-0.368	0.336
	12.5 %	-0.474	0.491	-0.361	0.323
	Europarl (same size as Bible)	-0.481	0.516	-0.365	0.333
Indonesian	Bible 100.0 %	-0.481	0.518	-0.371	0.343
Modern Greek	"	-0.479	0.514	-0.369	0.342
Russian	"	-0.482	0.520	-0.370	0.342

Table 2. Linear coefficients for each language; and for different sizes and different genres in English.

The interesting fact that comes into light is not so much the fact that the relation between size and saturation of analogical grids be a log–log relation, but the fact that it exhibits very similar slopes in all four languages. A reasonable explanation is that these coefficients are independent of the language because they characterize the corpus used. The corpus is defined by its size and its genre.

We first inquired whether the coefficients depend on the size of the corpus used. We performed the same experiment in English and let the size of the corpus vary: a half, a quarter, an eighth of the original size. The computation of the coefficients led to very similar results as shown in Table 2.

We then inquired the influence of the genre and performed the same experiment with the same size of text in English again. We chose the Europarl corpus for this experiment. Again, the computation of the linear coefficients led to very similar results, as shown in Table 2.

Further experiments with more parameters varying are required to confirm that the coefficients of the relationship between saturation and the size are always very similar. However, for the time being, we observe that the parameters are relatively close at least for these four languages whith different richness in morphology.

5 Conclusion

We studied analogical grids in different languages with different morphological richness. These analogical grids were automatically built from actual texts, using a technique which has been presented in previous work. Without surprise, languages known to be richer in morphology produce bigger and more analogical grids than languages less rich in morphology. Empty cells in such analogical grids are interesting because they could be filled by words that should then be tested against the actual language.

We studied the relation between size and saturation in analogical grids. Experimental results clearly showed that the logarithm of the saturation of an analogical grids linearly depends on the logarithm of its size. This is not so surprising. More interestingly, the computation of the coefficients characterizing this log-log linear relation led to the result that, across all the four languages used, and even when having size and genre varying in one language, these coefficients are almost always the same: the relation between the saturation and the size of an analogical grid would be almost independent of the size, the genre and the language of a text.

References

1. Beesley, K.R.: Consonant spreading in Arabic stems. In: Proceedings of COLING-ACL'98. vol. I, pp. 117–123. Montréal (Aug 1998), `http://www.aclweb.org/anthology/P98-1018`
2. Chan, E.: Structures and distributions in morphology learning. Ph.D. thesis, University of Pennsylvania. (2008), `http://nlp.cs.swarthmore.edu/~richardw/papers/chan2008-structures.pdf`
3. Dryer, M., Eisner, J.: Discovering morphological paradigms from plain text using a dirichlet process mixture model. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP'2011). pp. 616–627. Association for Computational Linguistics, Edinburgh, Scotland, UK (2011), `https://www.cs.jhu.edu/~jason/papers/dreyer+eisner.emnlp11.pdf`
4. Fam, R., Lepage, Y.: Morphological predictability of unseen words using computational analogy. In: Proceedings of the Computational Analogy Workshop at the 24th International Conference on Case-Based Reasoning (ICCBR-CA-16). pp. 51–60. Atlanta, Georgia (2016)
5. Goldsmith, J.: Unsupervised learning of the morphology of a natural language. Computational Linguistics 27, 153–198 (2001)
6. Hathout, N.: Acquisition of the morphological structure of the lexicon based on lexical similarity and formal analogy. In: Proceedings of the 3rd Textgraphs workshop on Graph-based Algorithms for Natural Language Processing. pp. 1–8. Coling 2008 Organizing Committee, Manchester, UK (August 2008), `http://www.aclweb.org/anthology/W08-2001`
7. Langlais, P., Yvon, F.: Scaling up analogical learning. In: Coling 2008: Companion volume: Posters. pp. 51–54. Coling 2008 Organizing Committee, Manchester, UK (August 2008), `http://www.aclweb.org/anthology/C08-2013`

8. Lepage, Y.: Analogies between binary images: Application to Chinese characters. In: Prade, H., Richard, G. (eds.) Computational Approaches to Analogical Reasoning: Current Trends, pp. 25–57. Springer, Berlin, Heidelberg (2014), http://dx.doi.org/10.1007/978-3-642-54516-0_2
9. Neuvel, S., Fulop, S.A.: Unsupervised learning of morphology without morphemes. In: Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning. pp. 31–40. Association for Computational Linguistics (July 2002), http://www.aclweb.org/anthology/W02-0604
10. Resnik, P., Olsen, M.B., Diab, M.: The Bible as a parallel corpus: Annotating the 'book of 2000 tongues'. Computers and the Humanities 33(1), 129–153 (1999), http://dx.doi.org/10.1023/A:1001798929185
11. Schone, P., Jurafsky, D.: Knowledge-free induction of morphology using latent semantic analysis. In: Proceedings of CoNLL-2000 and LLL-2000. pp. 67–72. Lisbon, Portugal (2000), http://web.stanford.edu/~jurafsky/W00-0712.pdf
12. Singh, R., Ford, A.: In praise of Sakatayana: some remarks on whole word morphology. In: Singh, R. (ed.) The Yearbook of South Asian Languages and Linguistics-200. Sage, Thousand Oaks (2000)
13. Stroppa, N., Yvon, F.: An analogical learner for morphological analysis. In: Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005). pp. 120–127. Association for Computational Linguistics, Ann Arbor, Michigan (June 2005), http://www.aclweb.org/anthology/W/W05/W05-0616
14. Wintner, S.: Natural Language Processing of Semitic Languages, chap. Morphological Processing of Semitic Languages, pp. 43–66. Springer, Berlin, Heidelberg (2014), http://dx.doi.org/10.1007/978-3-642-45358-8_2

Character–position arithmetic for analogy questions between word forms

Yves Lepage⋆

IPS, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, 808-0135 Fukuoka-ken, Japan
yves.lepage@waseda.jp

Abstract. We show how to answer analogy questions $A : B :: C : D$ of unknown D between word forms, by essentially relying on the basic arithmetic equality $D[i_B - i_A + i_C] = B[i_B] - A[i_A] + C[i_C]$ on characters and positions at the same time. We decompose the problem into two steps: specification and decoding. We examine several techniques to implement each of these two steps. We perform experiments on a set of positive and negative examples and assess the accuracy of combinations of techniques. We then evaluate the performance of the best combination of techniques on a large set of more than 40 million analogy questions from the training data of a shared task in morphology. We obtain the correct answer in 94 % of the cases.

Keywords: Formal analogy, analogy questions, character–position arithmetic.

1 Introduction

In this paper, we address the problem of answering analogy questions of the type $A : B :: C : D$ between word forms where the unknown is D. Our proposal consists in relying essentially on the intuitive basic arithmetic equality $D = B - A + C$. We propose to write this arithmetic equality using characters and positions at the same time:

$$D[i_B - i_A + i_C] = B[i_B] - A[i_A] + C[i_C] \tag{1}$$

The use of this arithmetic equality is directly inspired by the famous equality between vectors proposed in [7] to answer analogy questions between words in the framework of distributional semantics. This is now referred to as vector arithmetic and is always exemplified with:

$$\overrightarrow{\text{queen}} \approx \overrightarrow{\text{king}} - \overrightarrow{\text{man}} + \overrightarrow{\text{woman}} \qquad \text{(man : king :: woman : queen)} \tag{2}$$

⋆ This work was supported by a JSPS Grant, Number 15K00317 (Kakenhi C), entitled Language productivity: efficient extraction of productive analogical clusters and their evaluation using statistical machine translation.

This paper gives empirical support for the use of Equality (1) to answer analogy questions (which only involve commutation[1]) between word forms (not meaning). For relevance to morphology, no knowledge other than equality of characters is used, i.e., order of letters in an alphabet, or the like, is not used.

The rest of the paper is structured as follows. Section 2 shows how the answer to an analogy question can be specified by a character–position matrix computed using Equality (1). Specification refinements are introduced to solve problematic cases. Section 2 shows that decoding the answer from the character–position matrix can be viewed as an assignment problem and thus solved using a standard algorithm. Section 4 summarises the two previous sections by giving an algorithm for the proposed method. This proposed method is validated by two series of experiments in Section 5.

2 Specifying the answer of an analogy question

2.1 Known features of the answer to an analogy question

From previous research [4,10,2,3], it is known that the answer D to $A : B :: C : D$ is partially determined. In particular, its length, which characters it contains, and their number of occurrences, are known. In mathematical notations:

$$A : B :: C : D \quad \Rightarrow \begin{cases} |D| = |B| - |A| + |C| \\ |D|_c = |B|_c - |A|_c + |C|_c, \ \forall c \end{cases} \tag{3}$$

where $|X|$ stands for the length of string X and $|X|_c$ for the number of occurrences of character c in string X. The above equations are yet another instance of the general arithmetic equality $D = B - A + C$. In addition, some work [6] states that the LCS distance, noted d below, between the pair of terms is equal:

$$A : B :: C : D \quad \Rightarrow \begin{cases} d(A, B) = d(C, D) \\ d(A, C) = d(B, D) \end{cases} \tag{4}$$

2.2 Character–position arithmetic

In analogies of commutation, pieces are combined and exchanged to compose the four different terms of the analogy A, B, C and D. It is always possible to put some pieces in common to A and B in correspondence with some pieces in common to C and D (or the same by exchanging B and C in this statement). For instance, in

$$\text{hearty} : \text{unheartily} :: \text{lucky} : \text{unluckily}, \tag{5}$$

[1] See [5] or [6, middle of p. 161]. We exclude analogies of repetition, e.g., Indonesian *guru* : *guru-guru* :: *pelajar* : *pelajar-pelajar*; reduplication, e.g., Ancient Greek λύω : λέλυκα :: παύω : πέπαυκα; and mirroring, e.g., *abc* : *wxyz* :: *cba* : *zyxw*.

the pieces 'heart', common to A and B, and 'luck', common to C and D, correspond. They are exchanged on each side of the symbol ::. We can write $A[1\ldots5] = B[3\ldots7] = \text{heart}$ and $C[1\ldots4] = D[3\ldots6] = \text{luck}$; and further

$$D[3\ldots6] \;=\; \text{luck} \;=\; B[3\ldots7]-A[1\ldots5]+C[1\ldots4] \;=\; \text{heart}-\text{heart}+\text{luck},$$

where $3 = 3-1+1$ and $6 = 7-5+4$. This equality combines several instances of Equation (1) for several instances of indices (i_A, i_B, i_C, i_D): $(1,3,1,3)$, $(2,3,2,3)$, $\ldots$, $(5,7,4,6)$

To summarise the previous remarks, Proposition (6), which embeds Equality (1), can be laid as a hypothesis to be tested. Importantly, it only makes sense if either $A[i_A] = B[i_B]$ or $A[i_A] = C[i_C]$ holds, i.e., if either (i_A, i_B) or (i_A, i_C) are *match points* in the matrices representing $A : B$ and $A : C$ (see Figure 1).

$$A : B :: C : D \;\Rightarrow\; \forall i_D \in \mathbb{N} \;/\; 1 \le i_D \le |D|,\; \exists (i_A, i_B, i_C) \in \mathbb{N}^3 \;/\; \begin{cases} 1 \le i_A \le |A| \\ 1 \le i_B \le |B| \\ 1 \le i_C \le |C| \\ i_D = i_B - i_A + i_C \\ D[i_D] = B[i_B] - A[i_A] + C[i_C] \end{cases} \qquad (6)$$

2.3 Specification of the answer using a character–position matrix

Enforcing Proposition (6) allows to specify the solution of an analogy question as illustrated in Figure 1. For each index in the string D, all possible combinations of indices in A, B and C corresponding to match points in matrices $A : B$ and $A : C$ are examined and the number of instances of Equality (1) for that index in D is memorised in matrices $B : D$ and $C : D$. By adding up these values for each character that we know will appear in D, and for each index in D, a character–position matrix can be built, from which the answer can ultimately be decoded.

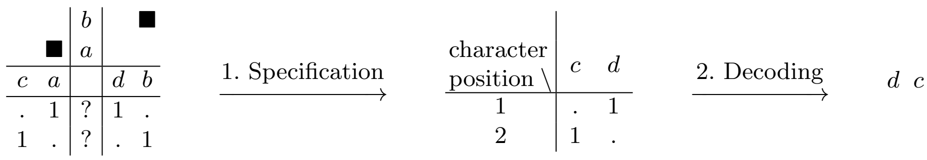


Fig. 1. Specification of the answer of the analogy question $ab : ac :: db : x$ using character–position arithmetic. The characters in the answer and their number of occurrences are known: $\{\, c : 1,\ d : 1\}$. The black squares (■) in the *upper* part on the *left* visualise the match points in the matrices $A : B$ and $A : C$. The *lower* part gives the number of instances of Equality (1) in each cell of matrices $B : D$ and $C : D$ using the match points and character–position arithmetic. In the *middle*, the character–position matrix summarises the evidence for each character and position (. stands for zero). This character–position matrix can be decoded into the answer dc on the *right*.

2.4 Virtual beginning and end match points

We now turn to a first problematic case where Proposition (6) is not verified, although it should be: work : sing :: you work : you sing. In this case, no triple of indices in A, B and C can be found for the first position in D corresponding to the character y. The character y in D can only come from the same character in C in the first position, as it does not appear in B. However, no character in A is equal to any character in B, so that there is no match point. So Proposition (6) does not hold.

Now, Proposition (6) holds for $\vdash$work$\dashv$: $\vdash$sing$\dashv$ :: $\vdash$you work$\dashv$: $\vdash$you sing$\dashv$ where beginning and end markers are added in the previous example. In that case, for position $i_D = 1$ in D corresponding to character y, the triple of indices $(i_A = 0, i_B = 0, i_C = 1)$ in A, B and C is such that: $D[1 = 0 - 0 + 1] = C[1] =$ y and $A[0] = B[0] = \vdash$. The addition of such markers is tantamount to the insertion of virtual match points in the matrices representing $A : B$ and $A : C$. On our set of positive examples (see Section 5), Proposition (6) holds for all examples when adding such virtual match points

2.5 Match points inside diagonal bands

We turn to a second problematic case for Proposition (6). Taking all possible match points into account may give too much weight for some of them, leading to wrong answers. This is the case for leaf : leaves :: wolf : D. Taking into account all possible match points in the matrices $A : B$ and $A : C$ leads to 10 instances of Equation (1) voting in favour of $D[1] =$ l, and only 7 in favour of $D[3] =$ l (while the situation is balanced for $D[1] =$ w and $D[3] =$ w with 8 equations each). This leads to the incorrect answer 'lowves' instead of 'wolves'.

The work in [4] considers only match points lying on the edit distance traces in $A : B$ and $A : C$, to create the answers to analogy questions. We follow this idea, but not as strictly. In [9, p. 106, illustrated on Figure 3], it is proven that the match points on edit distance traces between strings X and Y lie inside a diagonal band in the edit distance matrix. This diagonal band can be equivalently defined by Inequality (7) which uses the notion of similarity between two strings, i.e., the length of their longest common subsequence, noted s below, instead of the notion of edit distance.

$$-|X| + s(X,Y) \;\leq\; i_Y - i_X \;\leq\; |Y| - s(X,Y) \tag{7}$$

On the previous example, restricting to match points inside diagonal bands delimited by Inequality (7) in all matrices yields more instances of Equality (1) favouring $D[3] =$ l (4 equalities) over $D[1] =$ l (1 equality only), and no more equalities to support $D[3] =$ w; this leads to the correct answer 'wolves'.

2.6 Re-estimating values in the character–position matrix

We turn to a third and last problematic case for Proposition (6). For the analogy question in German *setzen* : *setzte* :: *lachen* : D, the character–position matrix built as described above somehow hesitates for the but last position between t and e: 2 instances of Equality (1) support $D[5] = t$, 6 support $D[5] = e$, 1 supports $D[6] = t$, and 4 support $D[6] = e$. This leads to the incorrect answer '*lachet*' instead of '*lachte*'.

The situation is similar to the one encountered in statistical machine translation where word-to-word correspondence probabilities should be re-estimated from the mere evidence that they appear in corresponding sentences. The answer consists in using the expectation–maximisation (EM) algorithm to estimate a probability distribution that will maximise the entropy over all possible word-to-word correspondences. The problem here is similar. Words in the source and target languages in machine translation correspond to character and positions in the character–position matrix. When applying the EM algorithm to the character–position matrix on the previous example, the probabilities for the previous characters and positions are re-estimated as follows: $p(D[5] = t) = 0.29$ now exceeds $p(D[5] = e) = 0.25$, and $p(D[6] = e) = 0.32$ exceeds $p(D[6] = t) = 0.28$. This leads to the correct solution '*lachte*'.

3 Decoding the answer of an analogy question

3.1 Solving an assignment problem

In the previous section, we showed how a character–position matrix can be built which assigns a probability to each character and position in the answer D of an analogy question $A : B :: C : D$. The final problem is thus an optimal assignment problem where each position should receive a character and each character should go to a position in D without conflict. This problem can be solved by the Hungarian method, or Kuhn's algorithm [1]. In our setting, we look for a solution of the assignment problem with a maximal cost.

It has also been shown that the Hungarian method is in fact the limit of an entropy maximisation problem [8]. So, we implement a naive and imperfect algorithm which works as follows. We assign each (possibly repeated) character to a position by scanning all characters in increasing order of entropy over all available positions. We assign a character to the position where it gets its highest probability. As several characters may have the same entropy simultaneously, we choose positions so as to avoid conflicts. If conflicts cannot be avoided, we simply stop the process and output no solution for the analogy question. Else, the characters and positions just assigned are removed, the entropies are computed again for the remaining characters and positions, and the process is repeated until all characters have been assigned to a position. If some remaining character cannot be assigned to any position, no solution is output.

This strategy is more prone to fail than the Hungarian method, and should thus be considered as a loose baseline.

3.2 Plurality of answers

There may be no answer, one answer or several answers to an analogy question. For instance, the analogy question $abcabc : gh :: mnkl : D$ has no solution; the analogy question easy : uneasy :: known : D has only one possible answer: $D =$ unknown; and the analogy question $aa : ab :: aaa : x$ has two possible answers only, aab or aba, if considered as an analogy of commutation (see Footnote 1).

From the solution delivered by the Hungarian method, it is possible to look in the character-position matrix and enumerate all other solutions of same cost by performing all possible exchanges between characters and positions.

4 Overview of the proposed method

4.1 Sketch of the method

Algorithm 1 Solving an analogy question $A : B :: C : x$.

```
def Solve(A, B, C):

    # 1. Specify the answer by a character-position matrix, M.
    ComputeKnownFeatures(A, B, C)
    M[i_D][c_D] = 0 for all c_D ∈ D and i_D ∈ {1,...,|D|}
    for each (i_A, i_B) / InsideDiagonal(i_A, i_B) and A[i_A] == B[i_B]:
        for each (i_C, i_D = i_B − i_A + i_C) / InsideAllDiagonals(i_A, i_B, i_C, i_D):
            M[i_D][C[i_C]] += 1
    #   Do same thing as above by exchanging B and C.
    ...
    M = ExpectationMaximisation(M)
    # 2. Decode the character-position matrix.
    list of pairs (i_D, c_D) = HungarianMethod(M)
    return EnumerateAllSolutions(M, list of pairs (i_D, c_D)))

def ComputeKnownFeatures(A, B, C):
    s(A,B), s(A,C) = similarity(A,B), similarity(A,C)
    s(B,D), s(C,D) = s(A,C) − |A| + |B|, s(A,B) − |A| + |C|
    |D| = |B| − |A| + |C|
    for each character c:
        occ♯_in_D[c] = occ♯_in_B[c] - occ♯_in_A[c] + occ♯_in_C[c]

def InsideDiagonal(i_X, i_Y):
    return −|X| + s(X,Y) ≤ i_Y − i_X ≤ |Y| − s(X,Y)

def InsideAllDiagonals(i_A, i_B, i_C, i_D):
    return all(InsideDiagonal(i_X,i_Y) for (X,Y) in [(A,C), (B,D), (C,D)])
```

Algorithm 1 sketches the method as already illustrated in Figure 1. After computing the features of the answer, a character–position matrix is built and its cells are filled using the character-position arithmetic. The values in the character–position matrix are then re-estimated using the expectation–maximisation algorithm. Decoding is performed using the Hungarian method. This outputs one answer. An enumeration of all character–position exchanges of same cost yields all possible answers.

4.2 Complexity analysis of the method

We give a very rough analysis of the complexity of the method. Computation of similarities or enumeration of the match points are basically square in the length of the strings, so that the most costly component in the algorithm is solving the assignment problem by the Hungarian method, known to be cubic in the size of the square matrix, i.e., cubic in the length of the solution in its best implementation. The convergence of the EM algorithm is difficult to estimate[2].

It is interesting to observe that the method is linear in the size of D in the best case, i.e., when $A = B$. In that case, the diagonal band is reduced to the main diagonal in the matrix $A : B$. Consequently, the character–position matrix exhibits a degenerated form where each character in C is assigned the same position in D as in C. Such a matrix is a degenerated case for the Hungarian method, which returns a solution in one pass over the matrix.

5 Experiments

To inspect the accuracy of our proposed method, we use an in-house data set of 160 examples, 113 positive examples and 47 negative examples. Most of the positive examples are from various languages: Arabic, Chinese, English, German, etc. They address complex phenomena, like parallel infixing, as in:

(German) *sprechen* : *ihr aussprächet* :: *nehmen* : *ihr ausnähmet*

In addition, some formal positive examples address incrementing problems:

$$abc : abcabc :: abcabcabc : abcabcabcabc$$
$$ab : aabb :: aaaaaabbbbbb : aaaaaaabbbbbbb$$

The purpose of the negative examples is to test the ability of our method not to deliver an incorrect answer. The negative examples are of the type:

$$ab : aabb :: aaabbb : aaabbbba$$

where the answer proposed, *aaabbbba*, is incorrect; the only correct answer, when restricting to analogies of commutation, is *aaaabbbb*. In this case, an algorithm that would blindly output all possible combinations of four a's and four b's in any order would have the incorrect answer in its set of solutions; it would thus fail the test.

We test different combinations of components: for the specification of the answer, use of all possible match points vs. only those inside diagonal bands

[2] We set the convergence threshold to the reciprocal of the number of cells in the character–position matrix, i.e., $1/|D|^2$. In general we observe convergence after a very small number of steps.

(Sect. 2.5), and use of the EM algorithm to re-estimate values in the character–position matrix vs. no use (Sect. 2.6); for decoding, use of the Hungarian method vs. our loose baseline (increasing entropies) (Sect. 3.1).

The results in Table 1 show that each of the components contributes to accuracy. Considering only match points inside diagonal bands allows to jump from below 66 % accuracy to almost 80 % and above. The EM algorithm may be of no utility or may add around 5 % in accuracy. As expected, the Hungarian method always beats our loose baseline by at least 5 % in accuracy. The best accuracy obtained is 91 % when decoding using the Hungarian method.

Match points inside diagonal band	EM algorithm	Decoding method	Recall	Precision	Accuracy
No	No	Increasing entropies	98	41	57
		Hungarian method	98	50	64
	Yes	Increasing entropies	98	41	57
		Hungarian method	98	53	66
Yes	No	Increasing entropies	99	70	78
		Hungarian method	99	82	87
	Yes	Increasing entropies	99	81	86
		Hungarian method	99	88	91

Table 1. Testing 8 different configurations to output the answer specified by character-position arithmetic. The best configuration is the last one.

Table 2 details the confusion matrix for the best configuration. Recall, precision and accuracy are computed in the standard way, and were reported in Table 1. This confusion matrix shows that the weakness of the method lies in too many negative predictions on positive examples. This is measured by a relatively low precision of 88 %.

	Positive predictions	Negative predictions	Total
Positive examples	TP = 99 (62 %)	FN = 14 (9 %)	113 (71 %)
Negative examples	FP = 1 (0 %)	TN = 46 (29 %)	47 (29 %)
Total	100 (63 %)	60 (37 %)	160 (100 %)

Table 2. Confusion matrix for the verification of analogies on 113 positive examples of analogies and 47 negative examples in the best configuration (see Table 1, last line).

This relatively low precision will now be nuanced by results on a much larger dataset which supposedly reflects more real analogy questions. This dataset is from Task 1 of Track 1 of SIGMORPHON 2016 Shared Task: Morphological Reinflection. We use all the offered languages: Arabic, Finnish, Georgian, German, Hungarian, Maltese, Navajo, Russian, Spanish and Turkish.[3] We extract analogy questions from such data by considering all analogies of form filtered by morphological features. For each analogy, four different analogy questions can be asked, each of the four terms becoming the answer.[4] As an example in Spanish, the four following questions correspond to the same analogy:

$$\begin{array}{lcl} \textit{alterado} : \textit{alterada} :: \textit{adeudados} : x & \Rightarrow & x = \textit{adeudadas} \\ \textit{alterada} : \textit{alterado} :: \textit{adeudadas} : x & \Rightarrow & x = \textit{adeudados} \\ \textit{adeudadas} : \textit{adeudados} :: \textit{alterada} : x & \Rightarrow & x = \textit{alterado} \\ \textit{adeudados} : \textit{adeudadas} :: \textit{alterado} : x & \Rightarrow & x = \textit{alterada} \end{array}$$

The number of analogy questions obtained in each language is given in Table 3. The total number over all languages exceeds 40 million analogy questions. For half of the languages, the percentage of correct answers is equal to or higher than 95 %. The total number of correct answers over all questions in each language reaches 94 %.

Language	Number of analogy questions	% of correct answers
Arabic	381,132	94
Finnish	3,076	95
Georgian	7,256,156	87
German	349,796	91
Hungarian	15,157,368	94
Maltese	10,000	97
Navajo	18,588,020	97
Russian	66,672	99
Spanish	95,564	95
Turkish	729,092	86
Total	42,636,876	94

Table 3. Solving analogy questions extracted from all training data of Task 1 of Track 1 from SIGMORPHON 2016 Shared Task for the best configuration of our proposed method (last line in Table 1)

6 Conclusion and future work

We showed how to answer analogy questions $A : B :: C : D$ of unknown D between strings of characters, by essentially relying on an intuitive basic arithmetic equality: $D[i_B - i_A + i_C] = B[i_B] - A[i_A] + C[i_C]$. We decomposed the problem into two steps: specification and decoding. We performed experiments on a set

[3] `https://github.com/ryancotterell/sigmorphon2016/tree/master/data/`. We use all the files of the type `<language>-task1-train`.

[4] This is not the task proposed in SIGMORPHON Shared Task, which consists in a machine learning task: predicting a word form given a lemma and morphological features after having learnt from the training data.

of positive and negative examples and measured the contribution of each of the components in accuracy. We further assessed the precision of the method on a very large set of more than 40 million analogy questions from the dataset of a shared task in morphology. We obtained the correct answer in 94 % of the cases.

As future direction, we want to carry on in testing the efficiency of the character–position arithmetic. For instance, it remains to inspect whether restricting further to those match points lying on edit distance traces helps or harms and whether we can dispense with the EM algorithm.

References

1. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistic Quarterly 2, 83—97 (1955), http://onlinelibrary.wiley.com/doi/10.1002/nav.3800020109/epdf
2. Langlais, P., Patry, A.: Translating unknown words by analogical learning. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). pp. 877–886 (2007), https://www.aclweb.org/anthology/D/D07/D07-1092.pdf
3. Langlais, P., Yvon, F.: Scaling up analogical learning. In: Proceedings of the 22nd International Conference on Computation Linguistics (Coling 2008). vol. Posters, pp. 51–54. Coling 2008 Organizing Committee, Manchester, UK (August 2008)
4. Lepage, Y.: Solving analogies on words: an algorithm. In: Proceedings of the Association for Computational Linguistics (ACL'98) and 17th International Conference on Computational Linguistics (COLING'98). vol. I, pp. 728–735. Montreal (Aug 1998), https://aclweb.org/anthology/P/P98/P98-1120.pdf
5. Lepage, Y.: De l'analogie rendant compte de la commutation en linguistique. Mémoire d'habilitation à diriger les recherches, Université de Grenoble (May 2003), https://tel.archives-ouvertes.fr/tel-00004372/document
6. Lepage, Y.: Proportional analogy in written language data. In: Gala, N., Rapp, R., Bel-Enguix, G. (eds.) Language, Production, Cognition and the Lexicon, pp. 1–23. Text, Speech and Language Technology 48, Springer International Publishing Switzerland (2014), http://link.springer.com/chapter/10.1007%2F978-3-319-08043-7_10
7. Mikolov, T., Yih, W.T., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013). pp. 746–751. Association for Computational Linguistics, Atlanta, Georgia (June 2013), http://www.aclweb.org/anthology/N13-1090
8. Sharify, M., Gaubert, S., Grigori, L.: Solution of the optimal assignment problem by diagonal scaling algorithms. ArXiv e-prints (Apr 2011), https://arxiv.org/pdf/1104.3830.pdf
9. Ukkonen, E.: Algorithms for approximate string matching. Information and Control 64, 100–118 (1985), http://www.sciencedirect.com/science/article/pii/S0019995885800462
10. Yvon, F., Stroppa, N., Miclet, L., Delhay, A.: Solving analogical equations on words. Technical report ENST2004D005, ENST (Jul 2004)

Analogical Localization: Flexible Plan Execution in Open Worlds

Scott Friedman, Mark Burstein, Jeffrey Rye, and Ugur Kuter
{friedman, burstein, rye, ukuter}@sift.net

SIFT, LLC
Minneapolis, MN, 55401, USA

Abstract

Cognitive systems face the challenge of pursuing changing goals in an open world with unpredictable collaborators and adversaries. Considerable work has focused on automated planning in dynamic worlds, and even re-planning and plan repair due to unexpected changes. Less work explores how humans and computers can negotiate to define shared goals and collaborate over the fulfillment of those goals. Our work takes a domain-general approach to *plan localization*, the problem of establishing the set of steps within the plan that are candidates (potentially after some adaptive repair actions) for next actions given the world's unforeseeen changes. We use analogical mapping to help agents determine the nearest states in a diverse plan relative to the current world state, identifying both the maximal *satisfied* states that the world conforms to presently, and the closest *desired* states adjacent to satisfied states that are both achievable by an action and makes progress toward the goal. These are demonstrated in a system called CLiC. The system's overall purpose is to engage in symmetric dialog with human users about goals and recommended actions to achieve those goals. Both the human and the system may choose to take those actions, or describe them to the other party. They may not always do what they are told. Preliminary results indicate that our approach suits collaborative situated agents with flexible goals in open worlds.

1 Introduction

Sometimes plans fail. The issues of uncertainty and exogenous events dominate, and require plan adaptions during execution of the plans. Progress in AI automated planning research has developed different approaches to plan adaptation [12, 10, 1, 14], but despite these advances, the problem of *analogically identifying* the most relevant goals and states in plans to achieve when a failure occurs in the world is still under-explored. This paper describes an approach to *plan localization* that addresses unexpected changes in the world, primarily due to well-intentioned collaborators.

Our approach is different than most work on reactive plan monitoring and repair. For instance, our system does not track whether the last action was precisely matched, since it exists in an open world without full knowledge of actions and paths to success. As other agents act in the world, it examines the world and reasons about the progress that has been made, if any. Our system identifies the plan states that are most *similar* to the world state, and uses analogical inferences to resume (or re-enter) the plan at the most preferred state. Our approach (1) computes a plan *locale*, (2) opportunistically identifies a *plan reentry state* if the locale is not a recognized future plan state, and (3) selects actions to enter to that re-entry state from the observed world state. For instance, in a block-building domain, the collaborator may have placed two or three blocks at once in an unexpected configuration, putting the world in an unexpected state, but the world state may analogically similar to a more desirable planned state (e.g., with slightly different block configuration), and we can change the world to satisfy that desirable state given some small action. Our approach enables the system to take opportunistic advantage of these situations and also take action despite setbacks.

By using analogical similarity matching to compare the world against future states in a diverse plan tree we can compute two different states of interest:

1. The best *satisfied state*: the state in the plan that is satisfied and closest to the goal, where distance ties are broken by structural similarity.
2. The nearest similar *desired state*: the plan state following a satisfied state that is most structurally similar to the current world state.

The satisfied and desired states comprise the *locale*, which is where the agent should focus its development of a next action to execute. Given the desired state found by this localization process, the system attempts to improvise actions— computed online via analogical inference— that will transition the world *back into the plan* by transitioning to the desired state. We call this the opportunistic *plan reentry*. Localization and reentry flexibly allow the agent to react to setbacks in the world as well as exploit unexpected world changes to more quickly achieve goals. This does not require a *complete* plan of all possible world states, but the more diverse and complete the plan is, the more resilient the localization behavior will be.

We have implemented this approach in CLiC, a dialogue-oriented agent that collaboratively builds block-based structures in a visual environment shared with a human collaborator. The high-level plan localization approach is illustrated in Figure 1 with a working example. The human collaborator suggests a goal structure in language, and CLiC uses its conceptual knowledge to envision that goal and any mentioned constraints. It then generates a diverse plan via regression to the current (empty table) state. CLiC both issues linguistic directives so that the user can move blocks, or alternatively, it responds to user directives by taking actions to move blocks on the table to incrementally achieve the shared goal. At each step, if the user responds unpredictably, e.g., by placing two blocks instead of one, or not placing them where directed, CLiC uses analogical plan localization to re-establish it's perceived position in the plan by matching against the shared world state. In this process it re-aligns which real blocks correspond to which planned blocks, and then proceeds to improvise new directives to move the plan forward, e.g., *"push block 8 together with block 6,"*.

We continue with a description of CLiC's planning, plan revision, and the task of collaborative building in Section 2. In Section 4, we describe our domain-general analogical plan localization approach, which we empirically support in Section 5 with multiple scenarios of CLiC utilizing this approach to flexibly mitigate and exploit unexpected changes in the world. We close with a brief discussion of related work (Section 6) and future work (Section 7).

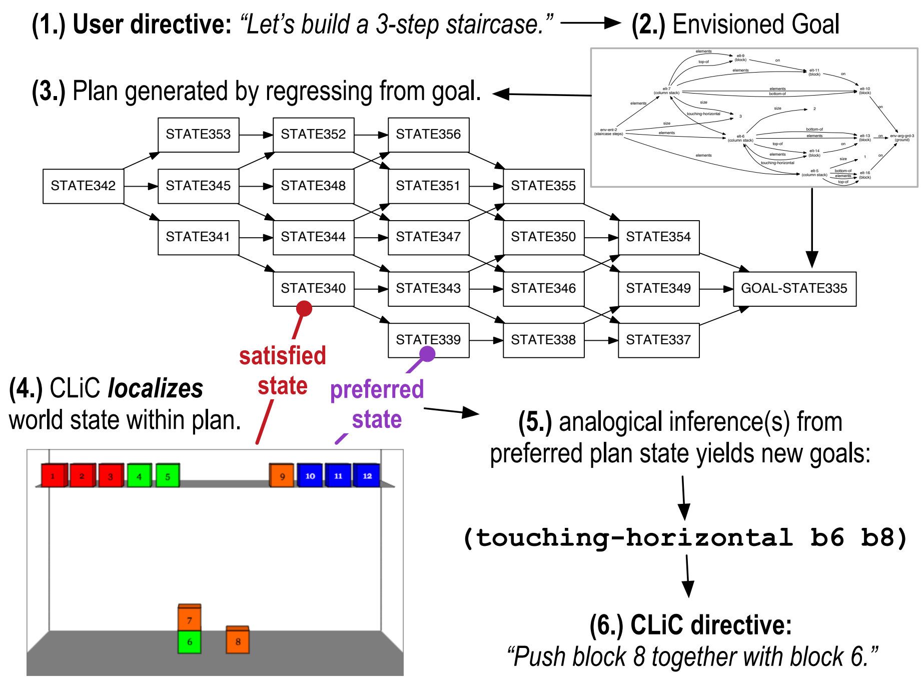


Fig. 1. CLiC localizes the unexpected world state within its plan via analogical mapping, and then selects an action via analogical inference. This allows it to re-enter the plan at a desirable state.

2 Background: CLiC & Collaborative Building

We are developing CLiC for DARPA's Communicating with Computers program, a program about contextually-grounded multi-modal communication with machines. We are presently working in two collaboration-oriented test domains: (1) collaborative goal selection and construction of structures using blocks and (2) a system for discussions with biologists about modeling and evaluating biochemical pathways. The basic CLiC reasoning infrastructure is shared across these domains. CLiC is a subsystem of a larger integration with the TRIPS agent-based dialog system architecture [4]. CLiC provides the domain reasoning for the dialogs occurring in both domains, and does planning and action selection, as well as responses to the user. Agents from the TRIPS system do the natural language understanding and maintain the state of the dialog as a high level goal tree, issuing high-level collaborative goals to CLiC.

CLiC reactively pursues collaborative goals by directly acting in the world or issuing linguistic directives to the user [11], and sometimes asking questions when more information is required to formulate a reasonable objective. In the biocuration domain, CLiC coordinates domain-specific biological modeling and reasoning agents, developed by other collaborators, to achieve collaborative goals with biologists. The rest of the paper focuses on the collaborative block-building setting, since CLiC's biocuration setting does not presently demand advance planning.

In the blocks world domain, the human collaborator suggests a named structure to build, e.g., row, stack, wall, staircase, with additional specifiers for size. CLiC uses conceptual knowledge to envision (as a set of logical and spatial relations over blocks) the form to be constructed. This is shown in Figure 2(a). When the user issues subsequent revisions to the goal, such as specifying sizes or colors— some of which may involve conflicting constraints— CLiC transforms these revisions into rules representing the specified constraints and then runs these rules to revise the goal. The result is illustrated in Figure 2(b), which is the same goal after being told *"the blocks should be green"* and then *"the tops should be red."*

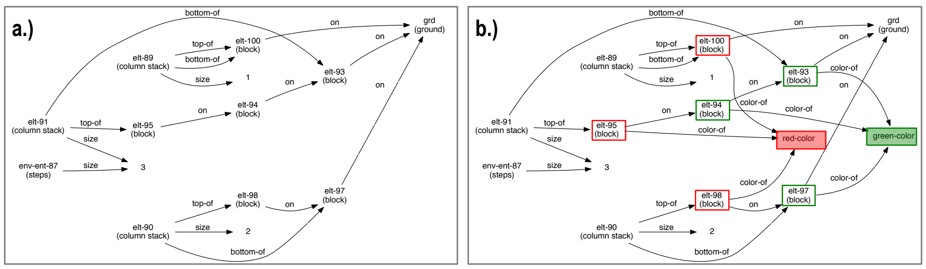


Fig. 2. Simplified semantic graph of a to-be-built staircase goal envisioned by CLiC, including a sequence of imaginary stacks with imaginary blocks: (a) the initial envisionment; (b) the envisionment after CLiC is told "the blocks should be green" and then "the tops should be red."

When CLiC receives a new goal, or when a goal is modified, it performs a simple plan regression search to generate a diverse plan from the solution state to the initial (empty table) state. One such plan graph CLiC generates to build a three-step staircase is shown in Figure 1. Unless specified by the user, the plan states are agnostic as to which *specific* blocks are used, what color they are, and the orienation of the structure (e.g., whether the staircase ascends to the right or left).

After agreeing on a goal— and in absence of additional directives and questions from the user— the rest of the block-building proceeds by (1) plan localization, (2) action selection, (3) action execution, and (4) awaiting collaborator actions before looping again. Before describing the plan localization approach that is our focus here, we briefly review the theory and implementation of CLiC's analogical reasoning.

3 Background: Analogy & Similarity

CLiC uses a domain-general implementation of the Structure-Mapping Theory [8] with a greedy algorithm similar to the Structure-Mapping Engine [5].

Structure-mapping takes two relational semantic graphs as input— a *base* and a *target*— and greedily computes a best *mapping* (i.e., nodes that correspond between base and target). Structure-mapping is a *maximal common subgraph* (MCS) problem, with the following hard constraints:

- **One-to-one**: a node in the base can match to at most one on the target, and the reverse. (This follows from the definition of MCS and isomorphism.)
- **Parallel connectivity**: two relational nodes can only correspond if their arguments also correspond.
- **Identicality**: relations (or attributes) can only correspond if their predicates (or categories) correspond.

Structure-mapping specifies an additional soft constraint:

- **Systematicity**: higher-order structures (e.g., relations over other relations or functions) are preferred over independent facts.

Structure-mapping computes a similarity score σ_b^t between the base b and target t, which is a weighted sum of nodes put into correspondence (higher-order relational nodes are weighted higher to implement systematicity). Intuitively, σ_b^t increases with the number of nodes in the mapping and with the systematicity of the nodes in the mapping, all else being equal. This is the objective function to maximize in the MCS solution. We use it to rank similarity: for a base b, target t_1 is more similar than target t_2 if $\sigma_b^{t_1} > \sigma_b^{t_2}$.

Finally, structure-mapping produces *analogical inferences* between base and target graphs. These analogical inferences are relations and attributes that are excluded from the mapping (i.e., they have no match in the MCS), that describe elements that *are* in the mapping. Analogical inferences can be projected from base-to-target or target-to-base. For example, suppose blocks in the world **`world-b1`** and **`world-b2`** correspond to blocks in a plan state **`plan-b1`** and **`plan-b2`**, respectively, and the relational statement **`(touching-horizontal plan-b1 plan-b2)`** is asserted in the plan, but the corresponding statement is *not* asserted in the world. Structure-mapping will produce the analogical inference **`(touching-horizontal world-b1 world-b2)`** as a projection in the world graph. Analogical inferences are relations or attributes projected whenever symbols correspond across graphs (e.g., ⟨`world-b1`, `plan-b1`⟩ and ⟨`world-b2`, `plan-b2`⟩) and one graph lacks a relation or attribute over the corresponding elements (e.g., **`(touching-horizontal world-b1 world-b2)`** is not asserted, so it is inferred). These inferences are not provably sound, but as we discuss next, they can be used very practically for improvising actions in the world.

4 Approach

CLiC's analogical plan localization runs whenever the world changes. For the purpose of illustration, suppose the task is to build a three-step staircase out of cubed blocks (comprised of six blocks in total) and CLiC is directing the human user how to build the structure. Suppose also that there is a single block **`b6`** on the table, and CLiC selects a planned action and tells the user, "Put block B6 on the table, and push B6 together with B8." This directive— should the human user obey it— would traverse a single edge in the plan graph and result in a transition to a planned state.

Now suppose that instead of putting **`b6`** next to **`b8`**, the user put **`b6`** on the table *apart* from **`b8`** and then immediately put **`b7`** on top of **`b6`**, as shown in Figure 1, bottom left. CLiC's plan localization runs after this unexpected (and undirected) world change.

The plan localization algorithm is given the world state w and sets its current state c to the initial state (e.g., **`STATE342`** in Figure 1), and then performs a best-first search:

1. If c is the goal state, the world satisfies the goal. Return success.
2. Otherwise, compute N as c's next immediate states in the plan graph.
3. Compute next state c' with highest similarity to the world: $c' = \max_{n \in N} \sigma_w^n$.
4. If the w to c' mapping has no analogical inferences to w, w satisfies state c' then set $c = c'$, loop to step 1.
5. Otherwise, return c as best satisfied state s_{sat} and return c' as desired state s_{des}.

This best-first search uses structural similarity as a guide through the state space of the plan to orient the agent within the plan. The agent has now approximated the best state in the plan whose conditions have been satisfied s_{sat} and a structurally similar state that might be opportunistically re-entered s_{des}.

The algorithm then computes actions that will change the world to satisfy s_{des} by computing analogical inferences in the mapping from w to s_{des}: all analogical inferences from s_{des} into w are treated as requirements to satisfy s_{des}. In the case described above (where **`b6`** is not touching **`b8`**), the system can force the world into **`STATE339`** in Figure 1 by achieving the analogical inference **`(touching-horizontal b6 b8)`**. This action will allow CLiC to jump form its previous locale with one block (i.e., **`STATE341`**) two steps ahead (i.e., to **`STATE339`**) by exploiting the unexpected change.

5 Experiments

We present four scenarios where CLiC localizes itself and achieves its goal, despite unexpected changes in the world and unexpected starting conditions.

5.1 Change spatial relations to reenter plan

In Section 4, we used the scenario in Figure 1 (on page 3) to outline the approach. In this scenario, the goal is to build a staircase. With a single block on the table, CLiC directs the human to put another block next to it; however, the user instead stacks two blocks on the table apart from the first. CLiC uses analogy to find the similar, desired plan state with a stack of two blocks and another block touching it on the table. The analogical inference is that the **`B6`** and **`B8`** should be touching, so it directs the user to push them together. This improvisation allows CLiC to reenter and complete the plan.

5.2 Utilize unexpected structure toward the goal

In this scenario, shown in Figure 3, the human suggests building a three-block stack. CLiC suggests stacking a second block on **`B1`**, but the human instead stacks **`B10`** **`B11`** apart from **`B1`**. CLiC then localizes this state to the penultimate state using the new **`B10`**/**`B11`** entities, and refocuses effort on the **`B10`**/**`B11`** stack instead of the initially-suggested **`B1`** stack, with the directive *“How about you put B12 on B11?”*

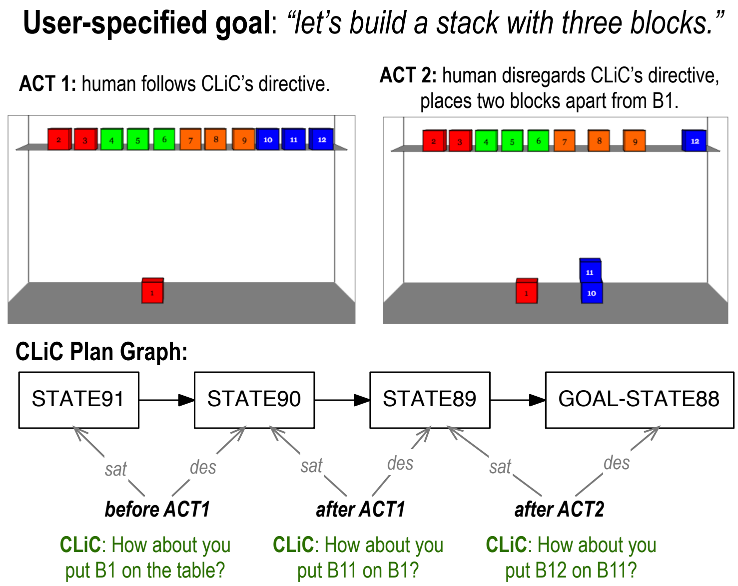


Fig. 3. CLiC exploits an unexpected change to refocus its effort toward the goal.

5.3 Reframe a symmetric goal with analogy

Staircases can be built in either direction, unless otherwise specified. In the Figure 4 scenario, CLiC and the human reach a state where a row of three blocks are on the table, and a fourth block is on top of the middle. From here, the staircase could still be built in either direction. CLiC suggests building in one direction, but the human disregards and puts a block on the other side.

CLiC reacts to this unexpected change by still localizing the world into the penultimate state and directing the user to put **`B3`** on **`B2`** to complete the staircase in the direction the collaborator determined.

5.4 Exploit starting conditions to enter a plan near the goal

In the scenario shown in Figure 5, the human places a four-block row on the table, with a fifth block on top, before specifying the staircase goal via dialogue. CLiC reactively localizes the already-developed world into its plan, identifying penultimate state **`STATE1988`** as the desired state s_{des}. The fourth block **`B4`** on the table corresponds to a second-row block in the planned state, so the analogical inferences include **`(touching-horizontal b10 b4)`** and **`(on b4 b3)`**. CLiC achieves these spatial relations by stacking **`B4`** on **`B3`** and then subsequently achieves the goal state.

This illustrates that analogical plan localization is useful for orienting the agent in unexpected starting states in addition to reconciling unexpected changes in the world.

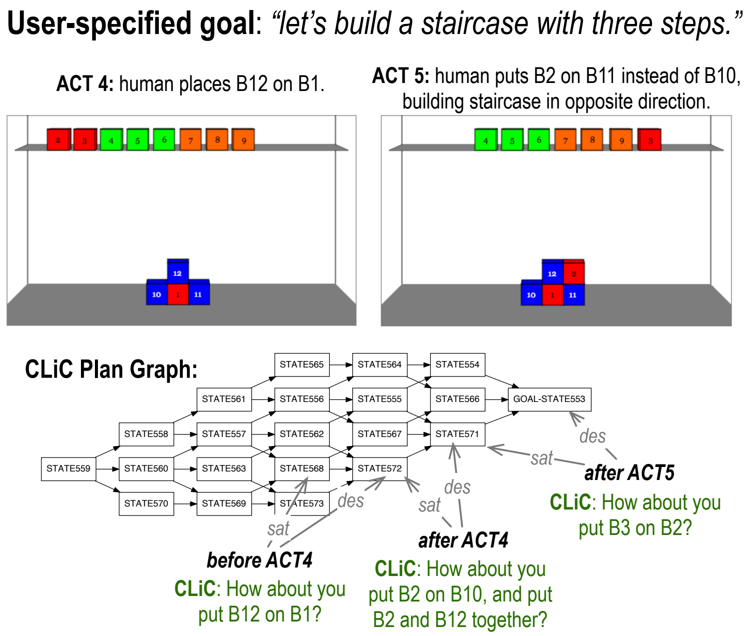


Fig. 4. CLiC reflects its goal staircase in the opposite direction to accommodate the collaborator.

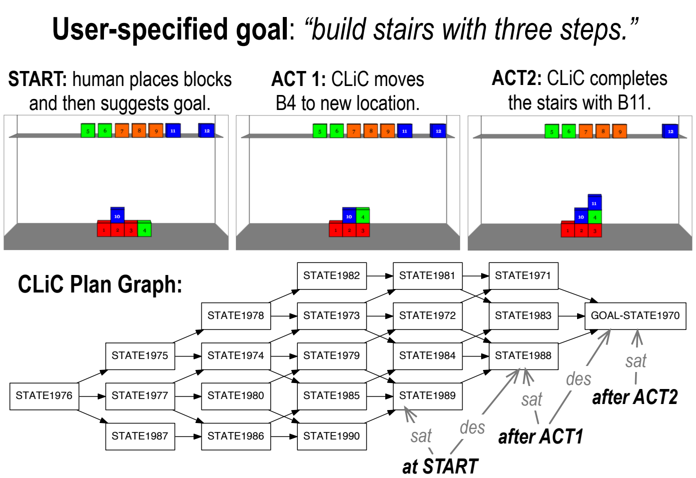


Fig. 5. CLiC adapts an existing scene via analogical inference to enter a plan near the goal state.

6 Related Work

Many AI planning systems have addressed re-planning, and plan repair. Decades of work on reactive planning (e.g., [12]) have investigated insertive and destructive approaches to reconciling the world state with plan states. The approach described in this paper differs from these AI reactive replanning and plan repair methods in that our approach uses structurally similar states to opportunistically repair plans, rather than just replanning from scratch given the current situation. As such it is somewhat similar to a case-based adaptation approach, though the adaptations are not made on prior cases, but on the initially developed plans for the goal.

Approaches for replanning from scratch [3] are fundamentally different than the approach described in this paper, since replanning generates new plans from the current state to the end of goals of a planning problems; conversely, our approach identifies opportunities for the agent to re-enter the plan, and then the existing plan can be reused.

AI plan repair typically focuses on locally repairing hierarchical plans when the system identifies a discrepancy during execution. For instance, replanning approaches like HOTRiDE [1] and SHOPLifter [10] are closely related to the case-based plan adaptation techniques proposed in the RepairSHOP system [14].

Approaches to diverse planning (e.g., [2, 13, 9]) have generated measurably diverse plans to cover more space of possible outcomes and explore possible state discrepancies in plans *a priori*, before execution. Among these are Coman & Munoz-Avila's work describing how to use case-based similarity/analogy methods in order to generate semantically-different plans. Unlike these approaches that use analogy and similarity to identify differences among diverse plans, our approach uses analogical reasoning to compare the world to the plan, orient the agent, and select actions.

7 Discussion & Future Work

We presented an approach to analogical plan localization that allows agents to flexibly recompute their locale of execution in a plan after drastic or unexpected world changes. In this setting, the plan is not considred as a strict policy for execution in the world; rather, it provides an ordering over partial world states that can be opportunistically reentered and traversed.

Our choice of analogy is particularly useful in a block-building domain. For instance, if our plan states that a blue block should go on a red block, it does not matter which *specific* blue block we choose. This means that analogical mapping can flexibly re-frame which *actual* blue block corresponds to the *planned* blue block in order to accommodate other entities and spatial relations in the analogical mapping. Other domains that allow substitution of entities will likewise benefit from this approach, whereas highly-specific goals are less amenable, e.g., if the goal in a logistics domain is for a specific truck to arrive with specific cargo at a specific location.

Our approach of localization via analogy is not limited to plans; we are using similar approaches to build systems that read articles and then localize extracted information within large models [7], akin to event recognition and information fusion. In this setting, extracted material– such as an abstract desription of an event– may localize against many concrete events in a large model, so we use a constrained similarity-based retrieval model similar to MAC/FAC [6].

Near-term future work includes validating this approach on other planning domains, as we believe that PDDL will support analogy. Other considerations include scaling to situations with larger branching factor, where CLiC's exhaustive regression planning is not tractable. In these cases, we could utilize HTN planning with diversity (e.g., [9]) to cover a subset of the plan space and support plan localization.

8 Acknowledgments

This work was supported in part by Contract W911NF-15-C-0238 with the US Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO). Approved for Public Release, Distribution Unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

1. Ayan, F., Kuter, U., Yaman, F., Goldman, R.P.: HOTRiDE: Hierarchical Ordered Task Replanning in Dynamic Environments. In: ICAPS-07 Workshop on Planning and Plan Execution for Real-World Systems (2007)
2. Coman, A., Muñoz-Avila, H.: Generating diverse plans using quantitative and qualitative plan distance metrics. In: Proceedings AAAI (2011)
3. Cushing, W., J.Benton, Kambhampati, S.: Replanning as deliberative re-selection of objectives. Tech. rep., Computer Science Department (2008)
4. Ferguson, G., Allen, J.: A cognitive model for collaborative agents. In: Proceedings of the AAAI 2011 Fall Symposium on Advances in Cognitive Systems. AAAI (2011)
5. Forbus, K.D., Ferguson, R.W., Lovett, A., Gentner, D.: Extending SME to handle large-scale cognitive modeling. Cognitive Science (2016)
6. Forbus, K.D., Gentner, D., Law, K.: MAC/FAC: A model of similarity-based retrieval. Cognitive science 19(2), 141–205 (1995)
7. Friedman, S., Burstein, M., McDonald, D., Paullada, A., Plotnick, A., Bobrow, R., Cochran, B., Pustejovsky, J., Anick, P.: Learning by reading: Extending and localizing against a model. In: Proceedings of the 4th Annual Conference on Advances in Cognitive Systems (2016)
8. Gentner, D.: Structure-mapping: A theoretical framework for analogy. Cognitive science 7(2), 155–170 (1983)
9. Goldman, R.P., Kuter, U.: Measuring plan diversity: Pathologies in existing approaches and a new plan distance metric. In: AAAI. pp. 3275–3282 (2015)
10. Kuter, U.: Dynamics of behavior and acting in dynamic environments: Forethought, reaction, and plan repair. Tech. Rep. 2012-1, SIFT (2012)
11. McDonald, D.D., Greenbacker, C.F.: 'If you've heard it, you can say it': towards an account of expressibility. In: Proceedings of 6th International NLG Conference. pp. 185–189 (2010)
12. Musliner, D.J., Durfee, E.H., Shin, K.G.: Execution monitoring and recovery planning with time. In: Proceedings of the Seventh IEEE Conference on Artificial Intelligence Applications. vol. 1, pp. 385–388. IEEE (1991)
13. Myers, K.L., Lee, T.J.: Generating qualitatively different plans through metatheoretic biases. In: AAAI. pp. 570–576. AAAI/MIT Press, Menlo Park, Cal. (July 1999)
14. Warfield, I., Hogg, C., Lee-Urban, S., Munoz-Avila, H.: Adaptation of hierarchical task network plans. In: FLAIRS-2007 (2007)

Towards a Domain-independent Method for Evaluating and Scoring Analogical Inferences

Joseph A Blass, Irina Rabkina, & Kenneth D. Forbus

Northwestern University, Evanston, IL, USA
{joeblass, irabkina}@u.northwestern.edu
forbus@northwestern.edu

Abstract. This paper proposes a domain-independent method to evaluate inferences for analogical reasoning, via a prototype system. The system assigns analogical inferences confidences based on the quality of the mapping and the system's confidence in the facts used to generate the inference. An initial implementation is applied to two domains.

Keywords: Analogical Reasoning, Inference Evaluation, Confidence.

1 Introduction

Any reasoning system which asserts facts through the processing and manipulation of previously known information ought to have a measure of confidence in the accuracy of those newly asserted facts. Even if a given reasoning technique is sound, inferred facts are only as accurate as the assumptions upon which they are based. For example, systems that reason via formal logic produce reliable inferences, but if the reasoning environment is complex enough, or a particular axiom is missing, contradictions may pass undetected. Furthermore, forward chaining systems overgenerate inferences, while backchaining systems are directed, but require a known goal for reasoning. On the other hand, probabilistic systems such as Bayes Nets [1] are good at determining how likely a particular inference is, but require a lot of training data or carefully hand-tuned priors. Analogy is a case-based reasoning technique that constructs an alignment between two cases, with a preference for shared structure, and uses that structure to make inferences from one case to another [2]. Inspired by human cognition, analogical reasoning does not require a fully articulated domain theory and can work from single examples and partial information. However, the inferences made by an analogical reasoning system may not be correct, and while there are evaluation measures based on the structure of the mapping and candidate inferences, all of the methods used in previous systems have been domain and/or task specific.

This paper proposes a unified approach to evaluating and scoring analogical inferences. It integrates logical reasoning, analogical reasoning, and probabilistic reasoning to provide confidence estimates for analogical inferences. We present an initial implementation and some experimental results as a proof of concept of these ideas.

1.1 SME, SAGE, and Cyc

The principles underlying our system are domain general. Our implementation uses the Structure-mapping Engine (SME, [3]) and a supplemented Cyc knowledge base [4]. What is important about the Cyc ontology for the present paper is that it provides *microtheories*. Microtheories serve as contexts, e.g. one microtheory might describe modern-day Chicago, while another describes Chicago as it was during the Fire. Microtheories can inherit from each other, e.g. when performing social reasoning, a common microtheory to include is HumanActivitiesMt, which as its name suggests, describes things people commonly do. Microtheories enable locally consistent reasoning, even though the knowledge base (KB) taken as a whole is inconsistent, e.g. there are microtheories describing different, incompatible fictional worlds. For analogical reasoning, we implement cases as microtheories, which enables reasoning to be done with respect to different cases locally. All reasoning is done with respect to a *context*, that is, a microtheory and all of the microtheories it inherits from.

SME [3] is a computational model of analogy that computes mappings between two structured cases, a base and a target. Each mapping includes correspondences between elements in the two cases, candidate inferences based on those correspondences, and a structural evaluation score calculated based on the structural similarity between the two cases. The higher the score, the more similar the cases and the more trusted the mapping. The Sequential Analogical Generalization Engine (SAGE [5]) uses SME mappings to create generalizations between cases. These generalizations can then be used as cases for further SME comparisons. Rather than keep only facts common to all generalized cases, SAGE generalizations are a joint distribution over the facts in all constituent cases. Each fact is stored in the generalization together with its probability, that is, the proportion of cases in that generalization that contains it. Only facts whose probability falls below a preset threshold are not included in the generalization. This scheme allows the generalization to maintain information about which facts are likely, not only which are universal. For example, consider a generalization composed of three cases that describe dogs: a Golden Retriever, a yellow Labrador, and a Dalmatian. The generalization will have the fact that a dog has 4 legs with probability 1.0 and the fact that it has yellow fur with a probability of 0.67. The inference evaluation system makes use of these probabilities, along with the structural evaluation score.

2 Inference Evaluation

When the system reasons its way to a new fact in a context, it can either be certain it is true, certain it is false, or somewhere in between. The system uses disjointness reasoning, logical contradiction and implication, and the parameters of SME mappings to determine the system's confidence that an inference is true. All reasoning is done with respect to the context in which the inference is to be asserted.

2.1 Disjointness Reasoning, Contradiction, and Implication

If the system has inferred that an entity is of a certain type, and there is already a contextualized assertion that it is of another type that is by definition disjoint from the first, the system simply rejects that inference. For example, if Fluffy is a dog, it cannot assert that it is a cat unless it first retracts that it is a dog. In the Cyc knowledge base, certain collections are marked as disjoint collection types, such that if an entity is an instance of one of those types, it cannot be an instance of another. When our system detects that an inference is of the form `(in-Context ?context (isa ?entity ?newType))`, it gathers all the other declarations of that entity's type in context `?context`. If any of those other types are disjoint with `?newType`, then the system rejects the inference. Inferences can also be rejected if they are contradicted by known implication rules. If there is a rule of the form `A -> ~I`, where `I` is the analogical inference, and `A` is known to be true, the inference can be rejected. Similarly, if there is a rule of the form `I -> A`, and `A` is explicitly known to be false, the inference can be rejected.

Implication is similar: If there is a rule of the form `A -> I`, and `A` is true, the inference has been confirmed. Similarly, if there is the rule `~A -> I`, and `A` is known to be false. The confidence in the implied fact is a function of the confidence assigned to the facts used to imply it. Contrapositives of the rules for implication and contradiction are generated on-the-fly. We do not assume rules are sufficiently complete to generate all inferences generated by analogy. Even if they were, analogy would be useful for focusing logical reasoning. The system makes use of forward chaining in a targeted fashion, only for verification, which is more efficient than simply forward-chaining.

2.2 Inferences from Analogical Reasoning

When the system derives an inference using an analogical mapping, it may be able to directly prove or disprove it. Failing that, it is desirable to have a measure of the extent to which the inference is trusted. The normalized SME match score is one such signal. Another is the degree to which the facts the inference is based on (in the base and target cases) are trusted. If the base case is a SAGE generalization, then the fact probability in the generalization tells us how likely that fact is within that generalization. For a non-generalized case, the system does not know the extent to which the case itself is an outlier or whether any one fact in the case is core to the overall concept that the case encodes. Inferences from individual cases should be trusted less than high-probability generalization facts, since there is evidence from the generalization that the high-probability facts are more common.

Putting it all together, analogical inferences are assigned confidence scores thus:

$$\mathrm{P(Inference)} = MatchScore * BaseTrust * \prod_{t\,\epsilon\,target\ support} P(t)$$

The BaseTrust is as described above: If the base case is a SAGE generalization then:

$$\mathrm{BaseTrust} = \prod_{b\,\epsilon\,base\ support} P(b)$$

And otherwise it is set to the default normalizing value (currently, 0.7).

Given this formula, confidence scores are always on the interval (0, 1): normalized match scores are on that interval, and fact confidence scores are on the interval (0, 1]. Confidence cannot be zero since zero-confidence facts are simply suppressed, rather than asserted with confidence zero. Normalized match scores are a measure of the *degree* of overlap between cases, rather than the total amount of information being mapped from one case to another. These can be low but are never zero: for a mapping to be generated at all, there must be some degree of overlap.

We use a product, rather than, say, a sum, of these components to keep confidence on the (0,1) interval. Intuitively it makes sense that a fact inferred from many facts should be trusted less than one inferred from only a few (if we are equally uncertain of the supporting facts). The more facts used to support an inference, the greater the chance that one of them is false and that the inference is therefore invalid. If the confidence scores were allowed to be greater than one, then the confidence of inferences might become greater as we moved further out along inference chains.

The system uses a Truth Maintenance System, which has a single argument to mark a belief as in or out. This renders combining evidence from multiple arguments moot.

2.3 Implementation

In the current implementation, facts that are assumed (for example, the details of the case that is to be reasoned about) have a confidence of 1. Our inference evaluator first tries to determine whether an inference is contradicted or implied; if it fails, it checks whether that inference is from analogy and scores it appropriately, and otherwise, assigns it the default normalizing score. Contradiction and implication are handled using backward chaining from axioms in the knowledge base, using resource bounds.

In our implementation, all inferences are given a confidence score and a reason for that score. The reason is the facts and axioms that were used to generate the score. For implied facts, the score is the product of the confidences of the facts that imply it (because perhaps those antecedents are not trustworthy). Contradicted facts are currently simply rejected, although in future implementations they will be scored based on the likelihood of the facts used to reject them. Confidence scoring for analogical inferences is described above. SME mapping scores can be normalized in three different ways, all of which are on in the interval [0:1]. The base normalized score is a measure of how much of the base case is mapped in the mapping, that is, how much of the base case overlaps with the target case. If the target is much larger than the base but the base is highly alignable with a sub-set of the target, the base normalized score will be quite high even if the match score is low. The target normalized score is the corresponding measure for how much of the target case is mapped, and the normalized score is the average of the base and target normalization scores. The default is the average normal score. Base normalization tends to be used in recognition tasks, where covering the entire base is the criteria, whereas target normalization tends to be used in reasoning tasks, where finding precedents that can lead to inferences within a more complex situation is important.

3 Evaluation

We tested the confidence scoring and contradiction components of this initial implementation on two tasks: Analogical Chaining and Moral Reasoning. Analogical Chaining is a commonsense reasoning technique that elaborates a case description by repeated analogy to small cases called Common Sense Units (CSUs) [6]. These CSUs can be extracted automatically from natural language, and are thus easy to provide to the reasoning system. As analogical chaining uses analogical reasoning, it does not require a fully articulated domain theory or rules constructed by experts, can reason with partial knowledge, and can use the same case for prediction or explanation. Analogical chaining has been tested on questions from the Choice of Plausible Alternatives commonsense reasoning test (COPA, [7]). As analogical chaining asserts inferences by analogy, then asserts new inferences building on those previous inferences, it is very valuable to give it a measure of confidence in those inferences.

We examined the performance of the inference evaluation system on 11 COPA questions, whose internal representations were automatically extracted from the English text of the question using EA-NLU [8]. These questions were selected because they require repeated analogical inference (i.e., chaining) to solve. The system had a case library of around 50 cases it could retrieve and reason with. For every question tested, the confidence scores assigned to inferences were lower the further down they were along the inference chain; this means the inferences that enabled the system to answer the questions had lower confidence scores than the intermediate inferences used to infer them, reflecting the system's lower confidence the further out it went from established facts. Inference scores ranged from 0.02 (for an inference made only using facts from the COPA question itself) down to 1×10^{-6} (for an inference several steps removed from the question facts). All but three questions did not involve any dead-end reasoning: analogical chaining found the correct answer for those questions without exploring any fruitless inference chains. We will examine two cases that involved dead-end reasoning in detail.

One question asks: "The egg splattered. What was the cause of this?" The answers are "I dropped it" and "I boiled it." The system first hypothesized that the egg splattering was caused by some unknown violent impact, and assigned that inference a confidence score of 0.01 (low inference scores are discussed below). It then hypothesized, as an alternative explanation for the egg splattering, that the egg hit the floor. This did not involve the abstract impact from the first inference, but was based only on the question facts. However, the mapping had a lower match score than the first, so it was given a confidence of 0.0004. The system then pursued, in separate reasoning contexts, explanations for the first two inferences. In the system's case library was a case describing how an object was violently impacted when it was hit with a rock, so it hypothesized that perhaps the unknown impact on the egg was caused by a rock. Despite being based on a higher confidence inference (the first inference asserted, where p=0.01), this inference had a low match score and therefore resulted in a score of 2×10^{-6}. Finally, the system used a fourth case to explain the inference that the egg hit the floor by hypothesizing that it was dropped. Despite being based on a lower-confidence fact than the inference about the rock, this inference had a higher match score and thus received a confidence of 2×10^{-5}. While low, this score is still an order of magnitude higher than the dead-end hypothesis about the rock based on more highly-trusted initial inference.

Another question asks: "The truck crashed into the motorcycle on the bridge. What happens as a result?" The answers are "The motorcyclist died" and "The bridge collapsed". The automatically constructed question representations involve only one statement about a motorcycle (and no motorcyclist) but several statements about the crashing event (who was involved, where it happened, etc.). The system retrieves cases based on what is present in the case, so it began by reasoning about a familiar case involving a vehicle crashing. In that case the vehicle was an airplane, so the system first posited that perhaps the crash in question involved an airplane malfunctioning ($p= 8\times10^{-5}$). The system then retrieved a story about a child falling out of bed and crashing onto the floor. It used this case to posit that the crash was caused when the truck fell out of bed ($p = 9\times10^{-5}$). Building on the airplane inference, it hypothesized that the airplane lost power ($p=1\times10^{-6}$), then that the motorcycle lost power, (0.012), and finally, having exhausted its knowledge of crashes, that the motorcyclist dies ($p=2\times10^{-5}$). In this case the correct inference has a lower confidence score than all but one of the dead-end inferences.

This example illustrates the pitfalls both of Analogical Chaining and of the inference evaluation system. After the system had posited the airplane, it was all too happy to continue reasoning about it, and the match scores were high enough along that reasoning chain (and low enough for the case that gave it the answer) that those erroneous inferences were scored much higher than the one that seems obvious to humans (humans of course have much prior knowledge the system lacks). The system can be led astray and mask the utility of useful inferences if it marks even one incorrect inference as highly probable. Furthermore, it seems wrong to give the system a hard-and-fast rule stating that airplanes cannot be involved in car crashes. Such a situation may be extremely unlikely, which could be recognized by accumulating cases in a generalization about car and motorcycle crashes, but, as Hollywood has shown us, it's not impossible. This raises an important point about the interplay between analogical reasoning and first-principles reasoning. Analogical learning can provide explicit evidence of what can happen, because analogical generalizations provide structured, relational probabilistic representations of what has happened. But analogical learning only implicitly gathers evidence about what cannot happen. First principles reasoning is better at ruling out the kinds of things that are impossible (e.g. vehicles cannot fall out of beds because they cannot fit in them).

MoralDM is a computational moral reasoning system that makes decisions by analogy to moral generalizations [9,10]. In one experiment, generalizations are formed from cases that either involve the principle of double effect [11], or do not. This principle states that harm caused as a side effect of preventing a greater harm is morally acceptable, but not harm caused in order to prevent that greater harm. The canonical example illustrating this principle is that most people say it is morally acceptable to switch a trolley that will hit five people onto a side track where it will hit one person (double effect), but not to instead push someone in front of the trolley to save those same five people (not double effect). In these moral generalizations, the facts indicating whether the case involves double effect and which case-specific action should be taken have probability 1, whereas the facts specific to the case (whether it is a trolley or torpedo doing the harm, for example, how many people are hurt, or what the mechanisms are to save those people) have lower probabilities. We took the inferences made in reasoning about moral cases by analogy to these generalizations and checked them with the inference evaluator. This was both to test the generalization normalizing component and to get a sense of whether even highly trusted inferences have low scores (the mappings that generate these inferences have high unnormalized scores). While the scores for the high-confidence facts are still quite low (in all cases approximately 0.02), the scores for the low-confidence facts are much lower, corresponding to the lower proportion of constituent cases in which they appear. In the same mapping where the decision fact was scored at 0.02, for example, the fact about the form the harm took was scored at 0.005. This example demonstrates the utility of taking generalization fact probability into account: had these inferences been made by analogy to an ungeneralized case, the inference evaluator would have given them both scores of 0.014. Using generalization probabilities gives the system a means to assess different inferences from the same mapping.

4 Related Work

Most previous work on analogical inference validation has been domain specific. For example, Ouyang and Forbus used first principles reasoning within the physics problem solving domain to validate candidate inferences produced by SME [12]. While the validation improved their system's performance, a complete domain model had to be assumed. Similarly, Klenk and Forbus used a small set of hand-encoded heuristics to verify candidate inferences during transfer learning [13]. While these were not a complete domain model, the heuristics were specific to inferences that could be made in the test domain. While the system described in this paper allows for domain-specific verification (i.e. through implies statements), it is domain-general. Furthermore, unlike previous systems which rated an inference as true or false, the current system allows for intermediate rankings.

Similar intermediate rankings have been used to evaluate inferences derived by non-analogical reasoning systems. Examples include fuzzy logic networks [14], Bayesian Logic models (BLOG, [15]), and Markov Logic Networks (MLNs, [16]). By assigning a fuzzy truth space to antecedents, fuzzy logic networks are able to derive fuzzy truth values for inferred consequents. They allow for incomplete domain knowledge, but do require a handwritten set of rules. Fuzzy rules can be used in combination with data sampled in a particular space to rule in or out inferences made within that space ([17]). Fuzzy logic networks assign qualitative truth values (e.g. "mostly true") to inferences, rather than calculating a quantitative confidence measure.

BLOG models and MLNs calculate numerical probabilities for inferences. BLOG models do so by defining a probability distribution over a set of possible worlds determined by prewritten axioms. A Metropolis-Hastings Markov chain Monte Carlo approach can then be used to make inferences from the distribution [18]. Using MCMC increases the time and computational cost of inference scoring in these models. MLNs take a different approach: they define a Markov Network over a set of first-order logic sentences and constants, such that a node exists for each grounding of each predicate and a connecting feature exists for every possible grounding of each sentence [15]. Weights are assigned to these features based on the likelihoods of the sentences they describe. A probability distribution is then specified over the ground network. The structure of the network can be learned, given the sentences their possible groundings [19]. The disadvantage of MLNs is scaling: the network grows with additional predicates, as well as additional potential groundings. This also means that every potential grounding of every potential predicate must be present in the training set.

The presented inference evaluation technique could be used in other analogical reasoning systems that score (or could score) the quality of their matches (that is, which have a measure similar to SME's match score). For example, inferences in AMBR ([20]) have evidence accrued in favor and against them, based on semantic and structural similarity. Top scoring hypotheses are asserted into the reasoning environment, but the amount of evidence in favor of them is not. If this evidence were stored as a confidence measure of facts as they are asserted, future inferences could be made not only on the basis of evidence in favor or against them, but the degree to which that evidence is itself believed.

In HDTP (e.g. [21]), analogical mappings are constructed via a process of anti-unification. For example, a formula $p_1(a)$ in a base and $p_2(a)$ in a target is replaced in the mapping by a general predicate P(a), where P is a generalization of both predicates p_1 and p_2. A measure of similarity of P to p_1 and p_2 could be used to score inferences made using formula P(a); the scores of those inferences could then be used to score future inferences made using those inferred facts. In HDTP, inferences are checked for logical consistency; expanding logical consistency checks for inferences is the next extension to be performed on our system.

5 Future Work

Even mappings with high unnormalized match scores, indicating a high quality match, may have low normalized match scores, depending on the relative size of the cases and how much information is left out of the mapping. In the current implementation, low confidence scores assigned to analogical inferences were driven largely by low normalized mapping scores. Small cases with little structural overlap should yield low-confidence scores, since the mappings used to generate the inferences are not seen by the system as being particularly reliable, informative mappings (as indicated by the low score). However, while analogical inferences should have lower scores than logically implied inferences, they should not be vanishingly low. One possibility is to use the highest normalized score as a multiplier in calculating inference confidence scores, rather than always using the same mapping score normalizing function. Each function provides different information, but a high score in either indicates that the mapping includes a high degree of overlap from one case to another. Scoring inferences using the highest normalization score will still involve incorporating the score of the match, the score of the justifying target facts, and the probability of the generalization facts.

Given the ubiquity of certain role predicates (`objectActedOn`, `performedBy`, etc.) analogical chaining can make some inferences that, to a human, seem quite silly. Having the ability to rule out those silly inferences using logical forms of commonsense is desirable but is not being done in the current implementation. The Cyc knowledge base contains millions of axioms, but we are currently only using a small subset (the disjointWith axioms). We plan to explore reasoning techniques that enable us to exploit more of this knowledge, especially horn clauses and implication statements, for constraint-checking (e.g. [22]).

Contradictions should perhaps be asserted with a confidence proportional to the scores of the facts contradicting them, rather than suppressed entirely. If facts are seen as relatively likely, then the contradiction is also likely. If contradictions are asserted, they will must signal which facts contradict them, to keep reasoning consistent.

Many analogy inferences involve positing skolem entities. These are entities present in the base and participating in the candidate inference but which are not present in the target. For example, the event in which the egg was impacted in the above example was posited as a skolem variable. Fundamentally, however, these are open variables, and implication can help resolve them. Contradiction works in a similar way, but instead can only rule out resolutions: just because a rule says that a particular individual cannot fill a role does not mean that it says that no one can.

Finally, further testing is needed on a wider range of domains, as well as further empirical testing of the analogical inference confidence scoring. While we have verified the implication and contradiction through disjointness components of the inference evaluation system are functioning properly, these need to be empirically tested. We can thereafter examine accruing and weighting evidence for and against facts.

6 Conclusion

We presented an initial implementation of a system to evaluate analogical inferences, which have no guarantee of being correct. The system can identify certain inferences as being more likely than others, but further evaluation and extension of the system is needed. Nonetheless, this seems to be a promising direction for inference validation and assessment, and points towards a method for resolving skolem variables in analogical inferences.

7 Acknowledgements

This research was supported by the Socio-Cognitive Architectures for Adaptable Autonomous Systems Program of the Office of Naval Research, N00014-13-1-0470.

References

1. Pearl, J., Russell, S.: Bayesian Networks. UCLA Cognitive Systems Laboratory, Technical Report R-277, November 2000. In M.A. Arbib (Ed.), Handbook of Brain Theory and Neural Networks, Cambridge, MA: MIT Press, 157-160 (2003).
2. Gentner, D.: Structure-Mapping: A Theoretical Framework for Analogy. Cognitive Science, 7(2), 155-170. (1983).
3. Forbus, K. D., Ferguson, R. W., Lovett, A., Gentner, D: Extending SME to Handle Large-Scale Cognitive Modeling. Cognitive Science. (2016).
4. Lenat, D.: CYC: A large-scale investment in knowledge infrastructure. Comm. of ACM, 38(11), 33-38 (1995).
5. McLure, M. D., Friedman, S. E., Forbus, K. D.: Extending Analogical Generalization with Near-Misses. Procs. of the 29th AAAI Conf. on Artificial Intelligence, Austin, TX (2015).
6. Blass, J. A., Forbus K. D.: Analogical Chaining with Natural Language Instruction for Commonsense Reasoning. Procs. Of the 31st AAAI Conference on Artificial Intelligence. San Francisco, CA. pp. 4357-4363 (February, 2017).
7. Roemmele, M., Bejan, C. A., Gordon, A. S.: Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning (2011, March).
8. Tomai, E., & Forbus, K. D.: EA NLU: Practical Language Understanding for Cognitive Modeling. In FLAIRS Conference. (2009, March)
9. Dehghani, M., Tomai, E., Forbus, K. D., Klenk, M.: An Integrated Reasoning Approach to Moral Decision-Making. Procs. Of the 31st AAAI Conference on Artificial Intelligence. pp. 1280-1286 (2008, July).
10. Blass, J. A., Forbus, K. D: Moral Decision-Making by Analogy: Generalizations vs. Exemplars. Procs. of the 29th AAAI Conference on Artificial Intelligence Austin, TX. (2015).
11. Foot, P.: The Problem of Abortion and the Doctrine of Double Effect. Oxford Review 5, pp. 5-15. (1967).
12. Ouyang, T. Y., & Forbus, K. D.: Strategy variations in analogical problem solving. Procs. of the 21st AAAI Conference on Artificial Intelligence. pp. 446-451. (2006, July).
13. Klenk, M., & Forbus, K.: Analogical model formulation for transfer learning in AP Physics. Artificial intelligence, 173(18), pp. 1615-1638. (2009).
14. Zadeh, L. A.: Fuzzy logic. Computer, 21(4), pp. 83-93. (1988).
15. Milch B, Marthi B, Russell S.: BLOG: Relational modeling with unknown objects. ICML Workshop on Stat. Rel. Learning and its Connections to Other Fields. pp. 67-73. (2004).
16. Richardson, M., Domingos, P.: Markov Logic Networks. Mac. Learn, 62, 107-136. (2006).
17. Ughetto, L., Dubois, D., Prade, H.: Implicative and conjunctive fuzzy rules-A tool for reasoning from knowledge and examples. In AAAI/IAAI, pp. 214-219. (1999, July).
18. Milch, B., Russell, S.: General-Purpose MCMC inference over relational structures. Procs. of the 22nd Conference on Uncertainty in Artificial Intelligence. pp. 349-358. (2006, July).
19. Kok, S., Domingos, P.: Learning the structure of Markov logic networks. In Proceedings of the 22nd International Conference on Machine Learning. pp. 441-448. (2005, August).
20. Kokinov, B., Petrov, A: Integrating memory and reasoning in analogy-making: The AMBR model. The analogical mind: Perspectives from cognitive science, pp. 59-124. (2001).
21. Schwering, A., Krumnack, U., Kühnberger, K., Gust, H.: Analogical reasoning with SMT and HDTP. 2nd European Cog Sci Conference, Delphi, Greece. pp. 652-657. (2007).
22. Sharma, A. B., Forbus, K. D.: Automatic Extraction of Efficient Axiom Sets from Large Knowledge Bases. Procs. of the 27th AAAI Conf. on Artificial Intelligence. (2013, June).

A complexity based approach for solving Hofstadter's analogies

Pierre-Alexandre Murena[1], Jean-Louis Dessalles[1], and Antoine Cornuéjols[2]

[1] Télécom ParisTech - Université Paris Saclay,
46 rue Barrault, 75013 Paris, France,
last@telecom-paristech.fr
[2] UMR MIA-Paris
AgroParisTech, INRA, Université Paris Saclay,
5 rue Claude Bernard, 75005 Paris
antoine.cornuejols@agroparistech.fr

Abstract. Analogical reasoning is a central problem both for human cognition and for artificial learning. Many aspects of this problem remain unsolved, though, and analogical reasoning is still a difficult task for machines. In this paper, we consider the problem of analogical reasoning and assume that the relevance of a solution can be measured by the complexity of the analogy. This hypothesis is tested in a basic alphanumeric micro-world. In order to compute complexity, we present specifications for a prototype language used to describe analogies. A few elementary operators for this language are exposed, and their complexity is discussed both from a theoretical and practical point of view. We expose several alternative definitions of relevance in analogical reasoning and show how they are related to complexity.

Keywords: Analogy, Complexity, Relevance

1 Introduction

Analogical reasoning is a fundamental ability of human mind which consists in establishing a mapping between two domains based on common representations. Analogies are involved in particular in the use of metaphors, humour [11] and in scientific research [4]. It is also the key ability measured in IQ tests [16]. Although it is perceived as a very basic and natural task by human beings, transferring this ability to computers remains a challenging task, whether for detecting, understanding, evaluating or producing analogies. A typical analogy can be expressed as follows: *'b' is to 'a' what 'd' is to 'c'*, which will be written **a : b :: c : d**. This problem involves two *domains*, called *source domain* and *target domain*. The analogy is based on the pairing of the transformation **a : b** in the source domain and the transformation **c : d** in the target domain. Several models have been developed so far to cope with analogical reasoning, but they are based on complex modelings and huge computing power, which is not plausible from a cognitive point of view. For example, softwares such as Copycat [8] and its successor Metacat [14] explore the possible mappings between the two involved problems (source and target problems).

The question of relevance is central in analogical reasoning in the sense that it defines the quality of the considered mappings. Because infinitely-many common properties can be found between two objects, a relevance measure has to be found to disqualify properties of little interest [6]. Moreover, several criteria may be considered to measure relevance of a mapping: the number of common properties [18], the abstraction level of the shared properties, structural alignment [5], pragmatic centrality [10] or representational distortion [7].

Inspired by some previous works [3], [15], [1], we consider in this paper that relevance in analogical reasoning can be measured by description length and Kolmogorov complexity (which is its formal equivalent). We propose the principles for a new generative language which can be used to describe analogical problems. Although it is presented in the domain of Hofstadter's analogies (i.e. analogical problems in alphanumerical domain), its principles are general and could be used in several other contexts. The idea of such a language is similar to the idea developed by [17] in the context of sequence continuation. This language offers a strict general framework and offers a cognitively plausible and generative description of analogies.

2 Representation bias for Hofstadter's micro-world

2.1 Presentation of Hofstadter's problem and its variant

In order to study general properties of proportional analogy, Douglas Hofstadter introduced a micro-world made up of letter-strings [9]. The choice of such a micro-world is justified by its simplicity and the wide variety of typical analogical problems it covers. The base domain of Hofstadter's micro-world is the alphabet, in which letters are considered as Platonic objects, hence as abstract entities. Elementary universal concepts are defined relatively to strings of letters, such as *first*, *last*, *successor* and *predecessor*. To this domain is added a base of semantic constructs defined by Hofstadter: copy-groups, successor-groups and predecessor-groups [8]. The typical problem considered by Hofstadter in this micro-world is the

We consider a slightly modified version of Hofstadter's problem. Our modifications correspond to an extension of the micro-world.

First, we consider an additional base alphabet: the number alphabet. This alphabet adds an infinite number of elements to the problem but does not make the base problem more complicated. Furthermore, this addition encourages the use of user-defined base structures and raises the issue of transfer between different domains. In particular, the analogy equation **ABC : ABD :: 123 : x** seems very basic for a human mind while it corresponds to a change of representation from the world of letters to the world of numbers. Besides, the use of other base alphabets can be justified by some prior knowledge of the users: for instance, it can be thought that the problem **ABC : ABD :: QWE : x** will admit a simple solution for any system familiar with the English keyboard layout.

Secondly, we consider a mapping from numbers to any base alphabet. This operation was discarded by Hofstadter's rules but seems important to us. The problem **ABC : ABD :: ABBCCC : x** relies on a such a mapping: the string **ABBCCC** is naturally described as "*n-th letter of the alphabet repeated n times for* $n \in \{1, 2, 3\}$".

The third major difference between our approach and Hofstadter's original works lies in the consideration of descriptive groups. While Hofstadter's approach is merely descriptive, we adopt a generative formalism in which the way strings were formed is taken into account. The static description of *copy-groups*, *successor-groups* or *predecessor-groups* is replaced in our framework by methods such as *copy*, *succession* or *predecession*.

2.2 Complexity-based description

In this paper, we will focus on the resolution of analogy equations of the form **A : B :: C : x** where $\mathbf{x}$ is unknown. We submit that the solution of such an equation is given by $\mathbf{x} = \arg\min_{\mathbf{x}} C(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{x})$ where the function C corresponds to the (minimum) description length for the four terms. Such a hypothesis is related to the well-known philosophical principle of Occam's razor stating the best choice is the shortest.

A strict definition for the description length is offered by algorithmic theory of information with Kolmogorov complexity [13]. Basically, the complexity $C_{\mathcal{M}}(x)$ of a string x corresponds to the length (in bits) of the shortest program on a Universal Turing Machine (UTM) $\mathcal{M}$ that produces x.

In practice, this quantity is not calculable, hence only upper-bounds are used to estimate the complexity of an object. An upper-bound corresponds to a restricted choice of programs or equivalently to the choice of a limited Turing Machine. In this paper, we consider a particular machine by defining an elementary language. The language we develop is an *ad hoc* construction encoding a theory of the domain of interest.

We do not consider here *prefix codes*, ie. decodable codes in which no code word can be the prefix of another code word. To cope with decoding, we consider that the code is space-delimited, which means that costless delimiters are present in it. This idea is in use in the Morse code for example. Morse code encodes letters by sequences of dashes and dots (ie. with a binary alphabet). A full word is given by a succession of letters separated by short breaks. These breaks are not part of the Morse code but are used to indicate the transition from one letter to another. In such contexts, the delimiters are supposed to be processed by the physical layer of the system, hence to ensure a uniquely decodable code while having no influence on complexity.

2.3 A generative language

Based on the specifications listed above, we develop a prototype language designed to produce and solve analogies. We present here the global characteristics of our language.

As mentioned, a major difference between our perspective and Hofstadter's works is the generative point of view. Largely inspired by Leyton's theory of shapes [12], we consider a description of the process generating analogies rather than a description of the analogies themselves. Any string will result from a transformation of the base alphabet: for instance, **ABCDE** is perceived as the sequence of the first five letters in the alphabet and **ZYX** as the sequence of the first three letters in the reversed alphabet.

In order to integrate this sequential transformation of an original string, we consider that the machine has access to a one-dimensional discrete tape. At each time step, the machine writes on this tape or modifies the previously written string. Thus, the base operation consists in copying the alphabet onto the tape. Thus, the generative procedure consists in a sequence of operations read from left to right and separated by commas. The operations are applied one by one and refer to understandable manipulations. Even if any operation may be incorporated to the language, we will consider here only a restricted set of predefined transformations, called operators $\{\mathcal{O}_1, \mathcal{O}_2, \dots\}$. The complexity of an operator is independent of the operation it performs. An upper bound of this complexity is the rank of the operator in the list of operators. For instance, the complexity of operator $\mathcal{O}_1$ is equal to 0, no matter how complex the corresponding operation actually is.

Besides, the instruction `next_block` is used to move to the next term in the analogy definition. For the analogy **A : B :: C : D**, the order of the blocks is **A**, **B**, **C** and **D**.

The core of the language is the use of a triple memory: a long-term domain memory, a long-term operator memory and a short-term memory. A string or a new operator can be put into short-term memory by means of the instruction `let`. The short-term memory can be accessed with the key instruction `mem`.

More precise information on the exact grammar chosen for the language can be found as supplementary material on the authors' webpage.

2.4 Basic operators

The list of operators available for the language determines the bias of the machine. The more operators are given to the system, the more sophisticated the obtained expressions can be.

The most basic set of programs is empty: it corresponds to a system capable of giving letters one by one only. Such a system is sufficient in some contexts. Consider for example the real problem of learning declension in a language. In order to learn a declension, students learn by heart a single example and transfer the acquired knowledge to new words. This corresponds for instance to the analogy **rosa : rosam :: vita : vitam** for a simple Latin declension. This analogy is encoded by the following code:

```
let('r','o','s','a'), let('v','i','t','a'),
  let(?, next_block, ?, 'm'),
  mem, 0, mem, 2, next_block, mem, 0, mem, 1;
```

This program has to be interpreted as follows: In the first line, the groups `'rosa'` and `'vita'` are put in short-term memory. The second line defines a new operator which displays the argument, switches to the next block, displays the argument again and finally adds the character `'m'`. The third line retrieves the just-defined operation and applies it successively to the two words, also retrieved from memory.

In order to build effective descriptions for more complex systems, additional operators can be defined. A list of possible operators is given in table 1.

Name	Description	Example
`copy`	Repeats the group a given number of times. Equivalent of Hofstadter's *copy-group*.	`'a', copy, 4;` outputs `aaaa`
`sequence`	Outputs the sequence of the first n elements of the group. Equivalent of Hofstadter's *copy-group*.	`alphabet, sequence, 3;` outputs `abc`
`shift`	Shifts the subgroups of n positions.	`alphabet, shift, 3;` outputs `defg...yz`
`shift_circular`	Circular version of the `shift` operator	`alphabet, shift_circular, 3;` outputs `defg...yzabc`
`reverse`	Reverses the order of elements in a group.	`alphabet,sequence,3,reverse;` outputs `cba`
`find`	Searches all occurrences of a group given as parameter.	`'a','b','a',find,'a',copy,2;` outputs `aabaa`
`find`	Selects last group	`'a','b','a', last, copy, 2;` outputs `abaa`

Table 1. Example of operators used by the language.

2.5 Using memory

The strength of the proposed language lies in its use of a triple memory to access elements of different nature: a long-term domain memory $\mathcal{M}_d$ storing domain descriptions (e.g. alphabets), a long-term operator description $\mathcal{M}_o$ storing system procedures to modify objects, and a short-term memory storing temporary elements. Managing memory is of major importance when it comes to producing programs of minimal length.

The access to elements in long-term memories $\mathcal{M}_d$ and $\mathcal{M}_o$ is hidden in the language for simplicity purpose, but it cannot be ignored. The designation of support alphabets (`alphabet`, `numbers`, `utf8`, `qwerty-keyboard`...), hence of the domain, and the designation of operators (`copy`, `sequence`, `find`...) are treated as proper nouns to encapsulate an access to an ordered memory. The rank of entities in memory is a characteristics of the machine and cannot be changed.

The user is in charge of the management of short-term memory. Entities (operators or strings) are stored in memory with the `let` meta-operator and accessed with the `mem` meta-operator. For example, the instruction `let('a')` will store the generation of `a` but the string is not written on the band. It will be written only when invoked from memory. The short-term memory is organized as a stack (hence last-in first-out): the parameter given to the `mem` operator is the depth of the element in the stack.

Using short-term memory is not compulsory to describe a string: the language syntax does not prevent from repeating identical instructions. However, in a context of finding a minimal description (which is the purpose of our framework), using memory is an important way to pool identical entities.

3 Relevance of a solution

3.1 From language to code

The principles outlined in previous section form a simplified grammar for our generative language. They are not sufficient yet to calculate the complexity of an analogy. The missing step is the formation of a binary code from an instruction.

The basic idea we use to obtain an efficient code consists in using a positional code in lists. This code associates element 0 to the blank symbol, `0` to element 1 and increments of 1 bit at for each element (`0`, `1`, `00`, `01`, `10`...). Using this code, the complexity of the n-th element of a list is $\lceil \log_2 n \rceil$.

The global description of the language is organized as a list of lists: a word is designated by the path inside the sequence of lists. For instance, the code for the character `d` corresponds to the code of domain memory (`1`), alphabet (`0`) and `d` (`01`), hence `1,0,01`. The code is not self-delimited: the delimiter is the comma symbol and can delimit a blank symbol. For instance, the number 2 is encoded by `1,,00`. Because a language word corresponds necessarily to a tree leaf, the code is uniquely decodable.

The complexity of an instruction is determined from the corresponding code. We propose to consider that the complexity corresponds directly to the number of bits required in the code. For instance, the complexity of the character 2 will be the number of bits in `1,0,0`, hence $C(2) = 3$. The same reasoning is applied to any instruction, including complex instructions describing complete analogies.

A way to build a cognitively plausible language encoding would consist in evaluating the ordering based on human experiments. Such experiments will have to be made in future research.

3.2 Relevance of a description

Several acceptable instructions can generate a given string. For example, the string `abc` can be produced either by `alphabet, sequence, 3;` (instruction 1) or `'a','b','c';` (instruction 2) or `alphabet, sequence, 2, 'c';` (instruction 3). These three instructions do not seem equally satisfying from a human point of view. We submit that the difference in terms of relevance can be quantified by their description length. Using a specific code description, the description lengths for the three previous instructions are respectively $DL_1 = 8$, $DL_2 = 10$ and $DL_3 = 12$. In this example, it is clear that the instruction with minimal description length corresponds to the most relevant description of the string. As a first step of our reasoning, we state that *the most relevant generative description of a string is the description of minimal description length.* An upper-bound for the Kolmogorov complexity of a string is defined as the description length of the most relevant instruction which outputs the string of interest. Despite the huge restriction applied to a general UTM by the choice of our language, the complexity remains non computable: its computation requires an exploration of all instructions producing the string, hence of an infinite space. Several solutions can be adopted in order to build the optimal program. First, greedy approaches would impose a research bias by the mean of a locally optimal exploration of the space of programs. Additionally to this guided exploration of the space of programs, a resource-bounded research can be considered [2].

3.3 Relevance of a solution for an analogy equation

Using the version of Kolmogorov complexity obtained by our system as described above, it is possible to apply the minimum complexity decision rule.

In order to evaluate the way human beings react to analogy problems, we proposed an experiment with several Hofstadters analogy problems.

Participants were 68 (36 female), ages 16-72, from various social and educational backgrounds. Each participant was given a series of analogies. The series were in the same order for all participants, and some questions were repeated several times in the experiment. All analogies had in common the source transformation **ABC : ABD**. The main results are presented in Table 2.

The results presented in Table 2 confirm that in most cases the most chosen solution corresponds to a minimum of cognitive complexity. The complexity is calculated here using our small language and the coding rules exposed earlier. Its limits are visible with the two examples **ABC : ABD :: 135 : x** and **ABC : ABD :: 147 : x**. In these examples, the language fails at describing the progression of the sequence "two by two" (1-3-5-7) or "three by three" (1-4-7-10) which would decrease the overall complexity.

However, despite the simplicity of the language used to calculate the complexity, it is noticeable that the most frequent solution adopted by the users corresponds a complexity drop. This property is not verified with only two problems: for the problem **ABC : ABD :: 122333 : x**, the large value of the complexity in the most frequent case is due to the limitations of the language which fails at providing a compact description of the complete analogy because of a too rigid grammar. In the case of the analogy **ABC : ABD :: XYZ : x**, adding the circularity constraint has a cost in the language, while it seems to be a natural operation for human beings.

The experiment also reveals a major weakness of our modeling: The descriptions provided by our language are static and do not depend on the environment.

Problem	Solution	Propostion	Complexity
IJK	IJL	93%	37
	IJD	2.9%	38
BCA	BCB	49%	42
	BDA	43%	46
AABABC	AABABD	74%	33
	AACABD	12%	46
IJKLM	IJKLN	62%	40
	IJLLM	15%	41
123	124	96%	27
	123	3%	31
KJI	KJJ	37%	43
	LJI	32%	46
IJK	IJL	93%	37
	IJD	2.9%	38
135	136	63%	35
	137	8.9%	37
BCD	BCE	81%	35
	BDE	5.9%	44
IJJKKK	IJJLLL	40%	52
	IJJKKL	25%	53
XYZ	XYA	85%	40
	IJD	4.4%	34
122333	122444	40%	56
	122334	31%	49
RSSTTT	RSSUUU	41%	54
	RSSTTU	31%	55
IJJKKK	IJJLLL	41%	52
	IJD	28%	53
AABABC	AABABD	72%	33
	AACABD	12%	46
MRRJJJ	MRRJJK	28%	64
	MRRKKK	19%	65
147	148	69%	36
	1410	10%	38

Table 2. Main results of the survey. For each problem, only the two main solutions are presented, with their frequency and the corresponding complexity.

On the contrary, the variations of the average answering time and the changes in the answers (when a same problem is repeated at several places) indicates clearly that having faced similar structures in the past helps in solving a new analogy. Finally, the relative relevance of two solutions is not necessarily sufficient to explain human preference in this matter, though. For instance, on the first problem, a large majority of people choose the **IJL** answer despite the small complexity difference. This possible divergence is related to research biases which are not taken into account in our approach. This effect is particularly visible with the more difficult analogy equation **ABC : ABD :: AABABC : x**. Very few humans notice the structure **A-AB-ABC**, hence the corresponding solution **x** = **AABABCD**. However, the structure **A-AB-ABC** is perceived as more relevant when presented.

We have shown that complexity offers a criterion to compare two given solutions to an analogy equation. This sole property is not sufficient in practice to obtain an analogy solver. Since the space of solutions is infinite, additional hypotheses must be considered in order to restrict the exploration space.

4 Conclusion

In this paper, we proposed to interpret analogical reasoning as a complexity minimization problem and to solve an analogy equation by taking the solution minimizing total complexity. Our approach relies on a restricted Turing machine: we proposed basic rules defining a small language adapted to Hofstadter's analogies. The language has been chosen to be generative (hence consistent with Leyton's theory of shapes) and not self-delimited (which allows compression with unspecified parameters). We gave general principles governing such a language. The system is flexible in the choice of the operations that can be involved for the description of an analogy. This language is associated to a code directly used in the computation of complexity. We use this code to measure the relative relevance of descriptions for a same string and the global relevance of a solution to an analogy. We used this code to measure the complexity of several analogies and noticed that the minimum complexity solution corresponds in most cases to the most frequent solution given by human beings.

Although the considered case might seem restrictive, our approach applies on a wider range of problems. Humans often justify their analogies with a semantic description. We consider our developed language as such. Similar languages can be developed for other analogies. Several issues remain open. A future research would be to develop a system able to generate descriptions automatically, hence to solve analogy equations automatically. The question of the performance evaluation of an analogy solver remains open: our framework measures only the relevance of a single solution. Some work has to be done to offer either a theoretical measure of the global quality for an analogy solver or an experimental validation of its efficiency. Finally, a real investigation on an extension of this language to other domains is needed in order to conclude on its actual generalization properties.

Acknowledgments. This research is supported by the program Futur & Ruptures (Institut Mines Télécom).

References

1. Bayoudh, M., Prade, H., Richard, G.: Evaluation of analogical proportions through Kolmogorov complexity. Knowledge-Based Systems 29, 20–30 (2012)
2. Buhrman, H., Fortnow, L., Laplante, S.: Resource-Bounded Kolmogorov Complexity Revisited. SIAM J. Comput. 31(3), 887–905 (2001)
3. Cornuéjols, A., Ales-Bianchetti, J.: Analogy and induction : which (missing) link? In: Workshop "Advances in Analogy Research : Integration of Theory and Data from Cognitive, Computational and Neural Sciences". Sofia, Bulgaria (1998)
4. Dunbar, K.: Designing for science: Implications from everyday, classroom, and professional settings. chap. What scientific thinking reveals about the nature of cognition., pp. 115–140. Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers (2001)
5. Gentner, D., Markman, A.B.: Structure mapping in analogy and similarity. American psychologist 52(1), 45 (1997)
6. Goodman, N.: Problems and Projects. Indianapolis: Bobbs-Merrill (1972)
7. Hodgetts, C.J., Hahn, U., Chater, N.: Transformation and alignment in similarity. Cognition 113(1), 62–79 (2009)
8. Hofstadter, D.: The Copycat Project: An Experiment in Nondeterminism and Creative Analogies. AI Memo 755, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (1984)
9. Hofstadter, D., Mitchell, M.: Fluid concepts and creative analogies. chap. The Copycat Project: A Model of Mental Fluidity and Analogy-making, pp. 205–267. Basic Books, Inc., New York, NY, USA (1995)
10. Holyoak, K.J., Thagard, P.: Analogical Mapping by Constraint Satisfaction. Cognitive Science 13(3), 295–355 (1989)
11. Holyoak, K., Holyoak, K., Thagard, P.: Mental Leaps: Analogy in Creative Thought. A Bradford book, Bradford Books (1996)
12. Leyton, M.: A Generative Theory of Shape. Springer (2001)
13. Li, M., Vitanyi, P.M.: An Introduction to Kolmogorov Complexity and Its Applications. Springer Publishing Company, Incorporated, 3 edn. (2008)
14. Marshall, J.B.: Metacat: a self-watching cognitive architecture for analogy-making and high-level perception. In: In Proceedings of the 24th Annual Conference of the Cognitive Science Society. Citeseer (1999)
15. Prade, H., Richard, G.: Testing Analogical Proportions with Google using Kolmogorov Information Theory. In: FLAIRS Conference (2009)
16. Ragni, M., Neubert, S.: Solving Raven's IQ-tests: an AI and cognitive modeling approach. In: Proceedings of the 20th European Conference on Artificial Intelligence. pp. 666–671. IOS Press (2012)
17. Strannegård, C., Nizamani, A.R., Sjöberg, A., Engström, F.: Bounded Kolmogorov Complexity Based on Cognitive Models, pp. 130–139. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
18. Tversky, A.: Features of similarity. Psychological review 84(4), 327 (1977)

A Language of Case Differences

Fadi Badra

Université Paris 13, Sorbonne Paris Cité, LIMICS, (UMR_S 1142), F-93430
Sorbonne Universités, UPMC Univ Paris 06, UMR_S 1142, LIMICS, F-75006
INSERM, U1142, LIMICS, F-75006, Paris, France
badra@univ-paris13.fr

Abstract. This paper contributes to a line of research that consists in applying qualitative reasoning techniques to the formalization of the case-based inference, and in particular, to its adaptation phase. The importance of capturing case differences has long been acknowledged in adaptation research, but research is still needed to properly represent and reason upon case differences. Assuming that case differences can be expressed as a set of feature differences, we show that Category Theory can be used as a mathematical framework to design a qualitative language in which both case differences, similarity paths and adaptation rules can be represented and reasoned upon symbolically.

Keywords: case differences · qualitative modeling · similarity path · adaptation

1 Introduction

Qualitative modeling provides formalisms that focus on how people represent themselves and reason about dynamical systems. Qualitative representations partition continuous quantities, and turn them into entities that can be reasoned upon symbolically [8]. The case-based inference aims at finding a complete description of a target problem by transferring information from a set of past problem-solving episodes, called cases, that are indexed in memory. Adaptation is the part of this process that aims at modifying a retrieved case when it can not be reused as it is in the new situation. Previous work on applying qualitative modeling techniques to adaptation includes a qualitative representation of relationships between quantities (called *variations* in [3]), such that co-occurrences of variations (called *co-variations* in [4]) can be interpreted as qualitative proportionalities. These proportionalities have been shown in [4] to play a great role in different commonsense inferences, and in particular, it has been suggested that adaptation was essentially an "analogical jump" performed on such proportionalities.

In existing formalizations, adaptation is recognized as being part of the case-based reasoning cycle [1], but surprisingly, the adaptation step is not included in the case-based analogical inference [17]. A study of the literature shows the adaptation step is always performed *after* the analogical inference (*i.e.*, retrieval, mapping, and transfer) has taken place, and only aims at modifying its result.

Some adaptation methods such as critique-based adaptation [11], or conservative adaptation [14] are used to resolve inconsistencies in the reused source case, whereas others such as differential adaptation [9], case-based adaptation [6] or adaptation by reformulation [15] modify the reused source case in order to fit the requirements on the target case. One of the reasons why adaptation is left out of the case-base inference is that adaptation essentially consists in reasoning on the *differences* that exist between two cases. While the importance of capturing case differences has long been acknowleged in adaptation research (see for example [13], for a recent review), research is still needed to properly represent and reason on case differences.

Establishing a difference between two states is the result of a comparison process. Comparisons are qualitative judgements that play an important role in similarity assessment and in the analogical inference. According to [12], "a comparison assembles two elements in order to come up with a third term that will tell their relationship". Comparison involves three ideas: the *source* of the comparison, the *target* of the comparison (what the source is compared to), and their relationship. For example, one could compare a sheep (the source) to a goat (the target), on how they forage (their relationship): a sheep would graze, whereas goats are browsers. Comparisons are usually made with respect to a particular *feature* (or property), shared by the objects under comparison, and which can be measured, like the size, the weight, or the type of forage [21]. Some results even suggests that people use aggregated features inferred from the features of individual objects to compare collections of objects [20].

Assuming that case differences can be expressed as a set of differences in feature value, we show that Category Theory can be used as a mathematical framework to design a qualitative language in which both case differences, but also "horizontal" connections of variations (similarity paths), and "vertical" connections (adaptation rules) can be represented and reasoned upon symbolically.

The paper is organized as follows. The next section provides some preliminary definitions. Feature comparisons are modeled in Sec. 3 as labeled arrows, and formalized in Sec. 4 as morphisms of a category. Two constructions are made on such categories: products (Sec. 5), and paths (Sec.6). In Sec.7, comparisons are ordered by generality using a subsumption relation. Finally, Sec.8 concludes the paper.

2 Preliminaries

Category theory is the mathematical study of algebras of functions [2]. A category consists of a set of *objects* and a set of *arrows*. For each arrow f, there are given objects $\mathtt{dom}(f)$ and $\mathtt{cod}(f)$ called the *domain* and the *codomain* of f. We write $f : A \longrightarrow B$ to indicate that $\mathtt{dom}(f) = A$ and $\mathtt{cod}(f) = B$. For two arrows f and g such that $\mathtt{cod}(f) = \mathtt{dom}(g)$, there is a given arrow $g \circ f$ called the *composite* of f and g. For each object A, there is a given arrow $1_A : A \longrightarrow A$ called the *identity arrow* of A. Arrows satisfy the *associativity law*: $h \circ (g \circ f) = (h \circ g) \circ f$ for all $f : A \longrightarrow B$, $g : B \longrightarrow C$, and $h : C \longrightarrow D$.

Identity arrows verify $f \circ 1_A = 1_B \circ f = f$ for all $f : A \longrightarrow B$. An arrow $f : A \longrightarrow B$ is called an *isomorphism* if there is an arrow $g : B \longrightarrow A$ such that $g \circ f = 1_A$ and $f \circ g = 1_B$. A *groupoid* is a category in which every arrow is an isomorphism. The category **Rel** is the category where objects are sets and arrows are binary relations. The identity arrow on a set A is the identity relation: $1_A = \{(a, a) \in A \times A \mid a \in A\}$. Given $f \subseteq A \times B$ and $g \subseteq B \times C$, the composition $g \circ f$ is defined as: $(a, c) \in g \circ f$ iff $\exists b \in B \mid (a, b) \in f$ and $(b, c) \in g$. Categories are mathematical structures which underlying structure is a quiver, *i.e.*, a directed graph where loops and multiple arrows between two vertices are allowed, on which the definition of a category adds constraints on identity morphisms, associativity, and composition. A *path* in the graph of a category is a sequence $\xrightarrow{c_1}\xrightarrow{c_2} \dots \xrightarrow{c_n}$ of arrows of $\mathbf{C}$ such that for all i, $\mathtt{dom}(\xrightarrow{c_{i+1}}) = \mathtt{cod}(\xrightarrow{c_i})$ A *path category* (or free category) generated by a directed graph is the category where the objects are vertices, and arrows are paths between objects. A *functor* $F : \mathbf{C} \longrightarrow \mathbf{D}$ between two categories $\mathbf{C}$ and $\mathbf{D}$ is a mapping of objects to objects and arrows to arrows that preserves domain and codomains, identities, and composition: $F(f : A \longrightarrow B) = F(f) : F(A) \longrightarrow F(B)$, $F(1_A) = 1_{F(A)}$, and $F(g \circ f) = F(g) \circ F(f)$. The *product* $\mathbf{C} \times \mathbf{D}$ of two categories $\mathbf{C}$ and $\mathbf{D}$ is the category of pairs and arrows. Its objects have the form (C, D), for $C \in \mathbf{C}$ and $D \in \mathbf{D}$, and its arrows have the form $(f, g) : (C, D) \longrightarrow (C', D')$ for $f : C \longrightarrow D \in \mathbf{C}$ and $g : C' \longrightarrow D' \in \mathbf{D}$. Compositions and units are defined componentwise, *i.e.*, $(f', g') \circ (f, g) = (f' \circ f\,, g' \circ g)$, and $1_{\mathbf{C}\times\mathbf{D}} = (1_{\mathbf{C}}, 1_{\mathbf{D}})$.

3 Modeling Feature Differences

We are interested in modeling the comparison between two values of a same feature. In the following, the term *feature* denotes either a binary variable (*i.e.*, a variable which takes one of the two values 0 or 1), or a nominal variable (*i.e.*, a variable which takes nominal values, like the color), or a quantity (*i.e.*, a variable which take values on ordinal, interval, or ratio scales [18]). The term *feature space* denotes the set of values taken by a particular feature.

A straightforward way to represent a comparison from a source A to a target B is to trace an arrow from A to B and to label this arrow with a term that represents their relationship. For example, an arrow named $\mathtt{g} \rightarrow \mathtt{b}$ can be used to represent the relationship in which the forage differs from **g**(raze) to **b**(rowse) from source to target (Fig. 1). The distinction between the source and the target

$$A \xrightarrow{\mathtt{g} \rightarrow \mathtt{b}} B$$

Fig. 1: A comparison of two feature values represented by a labeled arrow.

of a comparison makes the process by essence directional. It can be noted that this remains true even if the underlying relation is symmetrical. To illustrate this, consider the symmetrical binary relation `brother`, which relates two people when they are brothers. For two brothers A and B, both $\texttt{brother}(A, B)$ and $\texttt{brother}(B, A)$ hold (by symmetry), but $A \xrightarrow{\texttt{brother}} B$ and $B \xrightarrow{\texttt{brother}} A$ represent two different comparisons.

When the source and the target of the comparison are values of a same feature space, the comparison relation is transitive: if A can be compared with B and B with C, then A can be compared with C [5]. Besides, the relation is invertible, by which we mean that it is not possible to compare an object A to an object B without also being able to "reverse the viewpoint" and compare B with A with another relationship (possibly the same). For example, if a sheep A can be compared to a goat B with the relationship $\texttt{g} \rightarrow \texttt{b}$ ($\texttt{g}$ stands for **g**raze, and $\texttt{b}$ for **b**rowse), then an inverse relationship $\texttt{b} \rightarrow \texttt{g}$ can be used to compare B to A. It can be noted that feature value comparisons constitute a special case among similarity relationships. In the general case, similarity relationships are neither transitive nor invertible. For example, if Ted went to the same school as John and John went to the same school as Mary, it does not entail that Ted went to the same school as Mary. Comparisons may also not be invertible in simili ("a tree is like a man") or metaphors ("love is a battlefield"): we might say "a man is like a tree", meaning that a man has roots, but not "a tree is like a man" [21].

4 Formalization

Feature spaces can be formalized as categories, which we will call *feature categories*. The objects are the values of the feature space, and arrows represent comparisons between these values. Category Theory seems to be a natural setting to represent such comparisons, since arrows (also called morphisms) are the main "building blocks" of categories as mathematical structures. The categorical notion of composition of arrows corresponds to the transitivity of the comparison relation. Besides, each object of a category must be related to itself by an identity arrow. So representing a feature space as a category requires to distinguish identity arrows from difference arrows. Identity arrows, like $\texttt{d} \rightarrow \texttt{d}$ or $=$, have the same object as origin and destination, and express commonalities. Difference arrows, like $\texttt{d} \rightarrow \texttt{m}$ of $<$, have different origin and destination objects, and express differences. As all arrows are invertible in a feature category, the obtained category is a groupoid.

For example, the category **Bin** (Fig. 2) represents the quantity space of Boolean values, by taking as objects the two Boolean values 1 (True) and 0 (False), and as arrows the possible comparisons between these values. Feature categories may also represent quantity spaces. For example, consider the category $\mathbf{C}_<$, in which objects are elements of $\mathbb{N}$, and there are three arrows $\xrightarrow{=}$, $\xrightarrow{<}$, and $\xrightarrow{>}$. The arrow $\xrightarrow{=}$ is the identity arrow that links every integer $x \in \mathbb{N}$ to itself. The arrow $\xrightarrow{<}$ (resp., $\xrightarrow{>}$) links two integers x and y whenever $x < y$ (resp., $x > y$). Every arrow $\xrightarrow{<}$ is invertible since $y > x$ holds whenever $x < y$. The category **Area** (Fig. 3) represents location areas of apartments. Its objects are

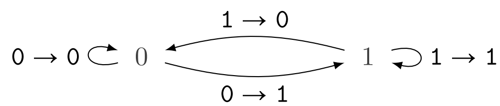


Fig. 2: The example category **Bin**, in which objects represent the Boolean values `0` and `1`, and arrows represent comparisons between these values.

the three nominal values **d**(owntown), **m**(idtown), and **u**(uptown), and its arrows the nine possible comparisons between them.

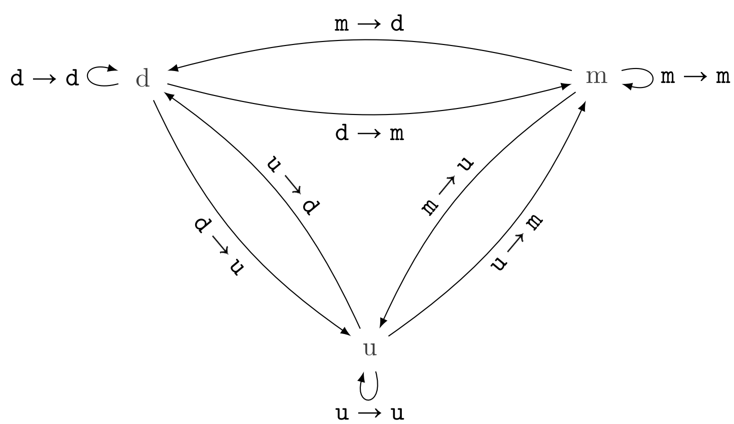


Fig. 3: The example category **Area**, in which objects represent the three areas `d`(*owntown*), `m`(*idtown*), and `u`(*uptown*), and arrows represent comparisons between areas.

4.1 Semantics

Feature categories are interpreted on a set (like a set of patients, of cooking recipes, etc.). Let $\mathcal{X}$ denote such a set. The semantics of a feature category $\mathbf{C}$ on a set $\mathcal{X}$ is given by a functor $.^{\mathcal{I}} : \mathbf{C} \longrightarrow \mathbf{Rel}$, called the *interpretation functor*, which maps each object of the category $\mathbf{C}$ to a subset of $\mathcal{X}$, and arrows to subsequent binary relations. The functor $.^{\mathcal{I}}$ generalizes the notion of binary variation. The definition of a binary variation as proposed in [3] corresponds to the indicator function of $.^{\mathcal{I}}$, when it is restricted to a given arrow of $\mathbf{C}$.

If there exists a *field* function $\varphi : \mathcal{X} \longrightarrow \mathbf{C}$, which maps each element of $\mathcal{X}$ to an object of $\mathbf{C}$, the interpretation functor $.^{\mathcal{I}}$ can be defined to map each object a of $\mathbf{C}$ to its inverse image by φ in $\mathcal{X}$, *i.e.*, to the set of elements of $\mathcal{X}$ which take the value a for the property φ:

$$a^{\mathcal{I}} = \{x \in \mathcal{X} \mid \varphi(x) = a\} \qquad \text{for an object } a \text{ of } \mathbf{C}$$
$$(a \to b)^{\mathcal{I}} = a^{\mathcal{I}} \times b^{\mathcal{I}} \subseteq \mathcal{X} \times \mathcal{X} \qquad \text{for an arrow } a \to b \text{ of } \mathbf{C}$$

For example, let $\mathcal{X}$ be a set of patients, and $\varphi : \mathcal{X} \longrightarrow \mathbf{C}_<$ be a field function that associates to each element of $\mathcal{X}$ an object of the category $\mathbf{C}_<$, representing the age of the patient. The interpretation functor $.^{\mathcal{I}} : \mathbf{C}_< \longrightarrow \mathbf{Rel}$ maps each age value $n \in \mathbb{N}$ to the set of patients having that age, and maps each comparison to the corresponding binary relation. The binary relation $(\xrightarrow{<})^{\mathcal{I}}$ is the set of pairs (a, b) of patients such that b is (strictly) older than a. Likewise, let $\mathcal{X}$ be a set of apartments, and $\varphi : \mathcal{X} \longrightarrow \mathbf{Area}$ be a field function that associates to each element of $\mathcal{X}$ an object of the category **Area**. The interpretation functor $.^{\mathcal{I}} : \mathbf{Area} \longrightarrow \mathbf{Rel}$ maps each nominal value to the set of apartments having the corresponding location area, and maps each comparison to the corresponding binary relation. The binary relation $(\xrightarrow{\mathtt{m \to d}})^{\mathcal{I}}$ is the set of pairs (a, b) of apartments such that a is located in **m**idtown and b is located in **d**owntown.

5 Representing Differences on Multiple Features

The *product* $\mathbf{C}_1 \times \mathbf{C}_2 \times \ldots \times \mathbf{C}_n$ of n comparison categories $\mathbf{C}_1$, $\mathbf{C}_2$,…,$\mathbf{C}_n$ has as objects the n-tuples $(a_1, a_2, \ldots, a_n)$ where a_i is an object of $\mathbf{C}_i$, and as arrows the n-tuples $(a_1 \to b_1, a_2 \to b_2, \ldots, a_n \to b_n)$, where $a_i \to b_i$ is an arrow of $\mathbf{C}_i$. For example, $(\xrightarrow{\mathtt{m \to d}}, \xrightarrow{=})$ is an arrow in the product $\mathbf{Area} \times \mathbf{C}_<$, and could be used to represent the comparison between an apartment located in **m**idtown and an apartment located in **d**owntown, both having the same price.

The interpretation functor $.^{\mathcal{I}}$ is extended to products in such a way that an element $x \in \mathcal{X}$ is in the interpretation of the product if it is common to all interpretations of $\mathbf{C}_i$'s:

$$(a_1, a_2, \ldots, a_n)^{\mathcal{I}} = \bigcap_i a_i{}^{\mathcal{I}} \qquad \text{for } n \text{ objects } a_i \text{ of } \mathbf{C}_i$$
$$(a_1 \to b_1, a_2 \to b_2, \ldots, a_n \to b_n)^{\mathcal{I}} = \bigcap_i (a_i \to b_i)^{\mathcal{I}} \qquad \text{for } n \text{ arrows } a_i \to b_i \text{ of } \mathbf{C}_i$$

6 Similarity

6.1 Analogy as Shared Differences

Two pairs are analogous when the same comparison can be made between them. When comparisons represent relations, this idea is consistent with the idea of analogy as a transfer of a relational structure, as outlined by Structure-mapping Theory [10]. For example, in the Andromeda galaxy, the X12 planets resolve around the X12 star, which can be represented as comparisons of the form "A:X12 planet $\xrightarrow{\text{resolve around}}$ X12 star". An analogy can be made between the

Andromeda galaxy and the solar system, by mapping these comparisons with comparisons such as "A:solar system planet $\xrightarrow{\text{resolve around}}$ sun". But the idea of analogy as shared comparisons can be generalized to the comparisons made to establish feature differences, that do *not* represent relations. For example, a same comparison $\mathtt{g} \to \mathtt{b}$ can be made from a sheep to a goat and from a cow to a moose: cow **g**raze, whereas moose **b**rowse. As a result, a cow is to a sheep what a moose is to a goat.

The same idea can be applied to logical proportions, which can be seen as shared comparisons. For two propositional variables x and y, there are four indicators: $I_1(x, y) = x \wedge y$, $I_2(x, y) = x \wedge \overline{y}$, $I_3(x, y) = \overline{x} \wedge y$, and $I_4(x, y) = \overline{x} \wedge \overline{y}$, and each logical proportion is defined by two distinct equivalences between these indicators [19]. For example, two pairs (x, y) and (z, t) are in *analogical proportion* if $I_2(x, y) \equiv I_2(z, t)$ and $I_3(x, y) \equiv I_3(z, t)$, *i.e.*, if $x \wedge \overline{y} \equiv z \wedge \overline{t}$ and $\overline{x} \wedge y \equiv \overline{z} \wedge t$ (here, $\equiv$ denotes the logical equivalence). Let $\mathbf{C}_x$, $\mathbf{C}_y$, $\mathbf{C}_z$, and $\mathbf{C}_t$ be the feature categories constructed as in Fig. 4. The category $\mathbf{C}_x$ contains the

$$\begin{tikzcd}
\overline{x} \arrow[loop left, "\overline{x} \to \overline{x}"] \arrow[r, bend right, "\overline{x} \to x"'] & x \arrow[l, bend right, "x \to \overline{x}"'] \arrow[loop right, "x \to x"]
\end{tikzcd}$$

Fig. 4: The category $\mathbf{C}_x$, with two objects (x and $\overline{x}$) for a propositional variable x, and arrows represent changes between these values.

two objects x and $\overline{x}$ for the propositional variable x. The interpretation functor $.^{\mathcal{I}}$ is defined using the valuation function v, which is a function from the set of propositional variables to $\{0, 1\}$, seen as the class of all subsets of a one-element set (0 is the empty set and 1 is the one-element set):

$$\begin{aligned}
a^{\mathcal{I}} &= v(a) \in \{0, 1\} && \text{for an object } a \text{ of } \mathbf{C}_x \\
(a \to b)^{\mathcal{I}} &= a^{\mathcal{I}} \times b^{\mathcal{I}} \subseteq \{0, 1\} \times \{0, 1\} && \text{for an arrow } a \to b \text{ of } \mathbf{C}_x
\end{aligned}$$

The arrow $(x \to \overline{x}, \overline{y} \to y)$ of the product $\mathbf{C}_x \times \mathbf{C}_y$ is interpreted as the binary relation $(x \to \overline{x}, \overline{y} \to y)^{\mathcal{I}} = v(x \wedge \overline{y}) \times v(\overline{x} \wedge y)$. Two pairs (x, y) and (z, t) are in analogical proportion if the interpretation of the two arrows $(x \to \overline{x}, \overline{y} \to y)$ and $(z \to \overline{z}, \overline{t} \to t)$ are the same, *i.e.*, if $(x \to \overline{x}, \overline{y} \to y)^{\mathcal{I}} = (z \to \overline{z}, \overline{t} \to t)^{\mathcal{I}}$. Likewise, two pairs (x, y) and (z, t) would be in paralogy if the interpretation of the arrows $(x \to \overline{x}, y \to \overline{y})$ and $(z \to \overline{z}, t \to \overline{t})$ are the same.

6.2 Similarity Paths

Let $\mathbf{C}$ be a feature category. A *similarity path* of $\mathbf{C}$ is a combination of arrows of $\mathbf{C}$. For example, $\xrightarrow{\mathtt{d} \to \mathtt{d}} \cdot \xrightarrow{\mathtt{d} \to \mathtt{m}}$ is a similarity path in the category **Area**. The free category $\mathcal{F}(\mathbf{C})$ generated by $\mathbf{C}$ is the category that has the paths of $\mathbf{C}$ as arrows. This definition can be extended to the product $\Pi = \mathbf{C}_1 \times \mathbf{C}_2 \times \ldots \times \mathbf{C}_n$ of n comparison categories $\mathbf{C}_1$, $\mathbf{C}_2$,...,$\mathbf{C}_n$. A path in Π is an arrow of the free category $\mathcal{F}(\Pi)$ generated by Π. For example, $(\xrightarrow{\texttt{d}\to\texttt{d}},\xrightarrow{<}) \cdot (\xrightarrow{\texttt{d}\to\texttt{m}},\xrightarrow{=})$ is a path in the free category generated by the product $\mathbf{Area} \times \mathbf{C}_<$.

The interpretation of a similarity path on the set $\mathcal{X}$ is given by the interpretation functor $.^{\mathcal{I}}$ which by definition of functors, preserves composition: $(\xrightarrow{c} \cdot \xrightarrow{d})^{\mathcal{I}} = (\xrightarrow{d})^{\mathcal{I}} \circ (\xrightarrow{c})^{\mathcal{I}}$. Here, the composition operation $\circ$ on the arrows of the category **Rel** is the usual composition of binary relations. This definition can also be extended to the product $\Pi = \mathbf{C}_1 \times \mathbf{C}_2 \times \ldots \times \mathbf{C}_n$ of n feature categories $\mathbf{C}_1$, $\mathbf{C}_2$,...,$\mathbf{C}_n$: for two sets of arrows $c_i, d_i \in \mathbf{C}_i$,

$$((\xrightarrow{c_1},\ldots,\xrightarrow{c_n}) \cdot (\xrightarrow{d_1},\ldots,\xrightarrow{d_n}))^{\mathcal{I}} = (\xrightarrow{d_1},\ldots,\xrightarrow{d_n})^{\mathcal{I}} \circ (\xrightarrow{c_1},\ldots,\xrightarrow{c_n})^{\mathcal{I}}$$

For example, for an apartment $\texttt{srce} \in \mathcal{X}$ located in **d**owntown, and an apartment $\texttt{tgt} \in \mathcal{X}$ located in **m**idtown, the pair $(\texttt{srce}, \texttt{tgt})$ is in the interpretation of the similarity path $(\xrightarrow{\texttt{d}\to\texttt{d}},\xrightarrow{<}) \cdot (\xrightarrow{\texttt{d}\to\texttt{m}},\xrightarrow{=})$ if there is an apartment $\texttt{pb}$ such that $\texttt{srce}\,(\xrightarrow{\texttt{d}\to\texttt{d}},\xrightarrow{<})^{\mathcal{I}}\,\texttt{pb}\,(\xrightarrow{\texttt{d}\to\texttt{m}},\xrightarrow{=})^{\mathcal{I}}\texttt{tgt}$, that is, such that the location of $\texttt{pb}$ is **d**owntown and its price is strictly greater than the price of $\texttt{srce}$, and equal to the price of $\texttt{tgt}$. This definition is consistent with the notion of similarity path, which is defined in [16] as a sequence of relations

$$\texttt{srce} = \texttt{pb}_0\, \texttt{r}_1\, \texttt{pb}_1\, \texttt{r}_2\, \texttt{pb}_2 \ldots \texttt{pb}_{q-1}\, \texttt{r}_q\, \texttt{pb}_q = \texttt{tgt}$$

such that the $\texttt{pb}_i$'s are problems and $\texttt{r}_i$'s are binary relations between problems.

7 Ordering Differences

7.1 A Subsumption Relation

A subsumption operator $\sqsubseteq$ enables to order comparisons by generality. Let $\mathbf{C}_1$ and $\mathbf{C}_2$ be two feature categories. For an arrow $\xrightarrow{c_1}$ of $\mathbf{C}_1$, and an arrow $\xrightarrow{c_2}$ of $\mathbf{C}_2$, we write $\xrightarrow{c_1} \sqsubseteq \xrightarrow{c_2}$ to represent that whenever an A can be compared to B using the comparison $\xrightarrow{c_1}$, then A can be compared to B using comparison $\xrightarrow{c_2}$. For example, in $\Pi = \mathbf{Area} \times \mathbf{C}_<$, the subsumption relation $\xrightarrow{\texttt{d}\to\texttt{m}} \sqsubseteq \xrightarrow{<}$ represents the fact that any apartment located in **d**owntown is more expensive than any apartment located in **m**idtown. The subsumption operator $\sqsubseteq$ can also relate the arrows of two product categories $\Pi_C = \mathbf{C}_1 \times \mathbf{C}_2 \times \ldots \times \mathbf{C}_k$ and $\Pi_D = \mathbf{D}_1 \times \mathbf{D}_2 \times \ldots \times \mathbf{D}_\ell$. For example, if $\Pi_C = \mathbf{C}_< \times \mathbf{Area}$ represents comparisons between the number of rooms and the location of apartments, and $\Pi_D = \mathbf{C}_<$ represents comparisons in price, then $(\xrightarrow{=},\xrightarrow{\texttt{m}\to\texttt{d}}) \sqsubseteq \xrightarrow{<}$ represents the fact that for a same number of rooms, an apartment located in **d**owntown is more expensive than an apartment located in **m**idtown.

Subsumption relations $\xrightarrow{c_1} \sqsubseteq \xrightarrow{c_2}$ are interpreted as set inclusions in $\mathcal{X} \times \mathcal{X}$:

$$\xrightarrow{c_1} \sqsubseteq \xrightarrow{c_2} \text{ if } \xrightarrow{c_1}^{\mathcal{I}} \subseteq \xrightarrow{c_2}^{\mathcal{I}}$$

This definition extends naturally to product categories:

$$(\xrightarrow{c_1}, \ldots, \xrightarrow{c_k}) \sqsubseteq (\xrightarrow{d_1}, \ldots, \xrightarrow{d_\ell}) \text{ if } (\xrightarrow{c_1}, \ldots, \xrightarrow{c_k})^{\mathcal{I}} \subseteq (\xrightarrow{d_1}, \ldots, \xrightarrow{d_\ell})^{\mathcal{I}}$$

A subsumption relation corresponds to the notion of *co-variation*, that is defined in [4] as a functional dependency between variations, and may be used to represent adaptation rules.

7.2 Analogical Jump

An analogical "jump" consists in making the hypothesis that a subsumption relation on comparisons holds for a given pair of objects. From a logical point of view, an analogical jump is defined in [7] as the following hypothetical rule of inference:

$$\text{if } P(x) = P(y) \text{ and } Q(x), \text{ then we can infer } Q(y)$$

For example, Bob's car and John's car share the property P of being a 1982 Mustang GLX V6 hatchbacks, and Bob's car has the property Q of having a price of 3500 \$. The inference is that the price of John's car should also be around 3500 \$. This schema can be rephrased using comparisons:

$$\text{from } x \xrightarrow{P} y, \text{ infer } x \xrightarrow{Q} y$$

In this schema, $\xrightarrow{P}$ and $\xrightarrow{Q}$ are two comparisons representing respectively that an element shares the property P with another element, and that it shares the property Q. This inference consists in making the hypothesis that the subsumption relation $\xrightarrow{P} \sqsubseteq \xrightarrow{Q}$ on comparisons holds for the pair (x, y). Such inference can also be made when the comparisons represent differences. For example, if $\Pi_C = \mathbf{C}_< \times \mathbf{Area}$ represents comparisons between the number of rooms and the location of apartments, and $\Pi_D = \mathbf{C}_<$ represents comparisons in price, then the subsumption relation $(\xrightarrow{=}, \xrightarrow{\mathtt{m} \to \mathtt{d}}) \sqsubseteq \xrightarrow{<}$ can be applied to a pair (x, y) of apartments to infer that an apartment y located in **d**owntown is more expensive than an apartment x with the same number of rooms, but located in **m**idtown.

8 Conclusion

Category Theory seems to be a natural setting to represent the feature comparisons made when establishing case differences. We showed that it can be used to and to design a qualitative language in which both case differences, similarity paths and adaptation rules can be represented and reasoned upon symbolically. We believe that such results open the way to new qualitative formalizations of the case-based inference, that would be able to integrate both retrieval and adaptation in a same analogical process.

References

1. Aamodt, A., Plaza, E.: Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun. 7, 39–59 (1994)
2. Adowey, S.: Category Theory. Oxford University Press (2010)
3. Badra, F.: Representing and Learning Variations. In: Int. Conf. Tools Artif. Intell. pp. 950–957. IEEE, Vietri sul Mare (2015)
4. Badra, F.: Reasoning with Co-variations. In: AIMSA. Varna, Bulgaria (2016)
5. Brown, R., Porter, T.: Category Theory: an abstract setting for analogy and comparison. In: What is Categ. Theory? Adv. Stud. Math. Log., pp. 257—-274. Polimetrica (2006)
6. Craw, S., Wiratunga, N., Rowe, R.C.: Learning adaptation knowledge to improve case-based reasoning. Artif. Intell. 170(16-17), 1175–1192 (nov 2006)
7. Davis, T.R., Russell, S.J.: A Logical Approach to Reasoning by Analogy. In: IJCAI Int. Jt. Conf. Artif. Intell. (1987)
8. Forbus, K.D.: Qualitative modeling. Wiley Interdiscip. Rev. Cogn. Sci. 2(4), 374–391 (2011)
9. Fuchs, B., Lieber, J., Mille, A., Napoli, A.: Differential adaptation: An operational approach to adaptation for solving numerical problems with CBR. Knowledge-Based Syst. 68, 103–114 (apr 2014)
10. Gentner, D.: Structure-Mapping: A Theoretical Framework for Analogy. Cogn. Sci. 7(2), 155–170 (1983)
11. Hammond, K.J.: CHEF: A Model of Case-based Planning. In: AAAI Proc. pp. 267–271 (1986)
12. Hoquet, T.: La liaison comme comparaison : sciences de rapports et logique de la relation. Astérion (2014), http://asterion.revues.org/2515
13. Jalali, V., Leake, D., Forouzandehmehr, N.: Ensemble of Adaptations for Classification: Learning Adaptation Rules for Categorical Features, pp. 186–202. Springer International Publishing, Cham (2016)
14. Lieber, J.: Application of the Revision Theory to Adaptation in Case-Based Reasoning: The Conservative Adaptation. In: ICCBR Proc. (2007)
15. Lieber, J., Napoli, A.: Correct and Complete Retrieval for Case-Based Problem-Solving. In: ECAI. pp. 68–72 (1998)
16. Melis, E., Lieber, J., Napoli, A.: Reformulation in case-based reasoning, pp. 172–183. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
17. Ontañón, S., Plaza, E.: On Knowledge Transfer in Case-Based Inference. In: Díaz-Agudo, B., Watson, I. (eds.) ICCBR Proc. pp. 312–326. Springer-Verlag Berlin Heidelberg (2012)
18. Paritosh, P.K.: Back of the Envelope Reasoning for Robust Quantitative Problem Solving Field of Computer Science. Tech. rep., Electrical Engineering and Computer Science Department, Northwestern University (2007)
19. Prade, H., Richard, G.: From analogical proportion to logical proportions: A survey. Stud. Comput. Intell. 548, 217–244 (2014)
20. Scontras, G., Graff, P., Goodman, N.D.: Comparing pluralities. Cognition 123(1), 190–197 (2012)
21. Tversky, A.: Features of Similarity. In: Readings Cogn. Sci. A Perspect. from Psychol. Artif. Intell., pp. 290–302. Elsevier Inc. (oct 2013)

A discussion of analogical-proportion based inference

Henri Prade[1,2] and Gilles Richard[1]

1. IRIT, Toulouse University, France
2. QCIS, University of Technology, Sydney, Australia
{prade, richard}@irit.fr,

Abstract. The Boolean expression of an analogical proportion, i.e., a statement of the form "a is to b as c is to d", expresses that "a differs from b as c differs from d, and vice-versa. This is the basis of an analogical inference principle, which is shown to be a particular instance of the analogical "jump": from $P(s)$, $P(t)$, and $Q(s)$, deduce $Q(t)$. Roughly speaking, an analogical proportion sounds like a sort of qualitative derivative. A counterpart of a first order Taylor-like formula indeed exists for affine Boolean functions. Affine functions can be predicted without error by means of analogical proportions. These affine functions are essentially the constants, the projections, the xor-based functions, and their complements. We discuss how one might take advantage of this state of fact for refining the scope of application of the analogical-proportion based inference to subparts of a Boolean function that may be assumed to be "locally" linear.

1 Introduction

Analogical proportions are statements of the form "a is to b as c is to d" that have been introduced at the time of Aristotle by mimicking numerical proportions. Such statements are appealing since they relate comparisons inside pair (a, b) to comparisons inside pair (c, d), by suggesting that "a differs from b in the same way as c differs from d", and for symmetry reason that "b differs from a in the same way as d differs from c".

Following a series of works aiming at formalizing the idea of analogical proportion in different settings, a Boolean logic modeling has been proposed almost a decade ago. This modeling formally acknowledges the above intuitive reading of an analogical proportion. The analogical-proportion based inference amounts to postulating that if analogical proportions hold on a series of features used to describe four situations a, b, c, d, such a proportion may also hold for other related attributes as well.

It turns out that such a view has been proved to be quite effective for classification tasks in particular. A natural question is then to try to understand why and in what respect. This question is not straightforward. An interesting clue has been recently obtained when showing that if (and only if) the classification function is an affine Boolean function, then the analogical-proportion based inference always predicts the right class [5]. This confirms previous experimental observations.

It also echoes some informal remarks pointing out the fact that an analogical proportion may be reminiscent of a qualitative notion of derivative. Indeed affine Boolean functions satisfy a first order Taylor-like formula as recalled in this paper. Since any Boolean function is piecewise linear (in terms of affine Boolean functions), one may wonder if one cannot take advantage of these facts for a better adjustment of the scope of the analogical-proportion based inference inside areas where the classification functions would be presumably linear.

The Boolean logic modeling of analogical proportions provides a simple basis for computing the result of an analogical inference. The paper mainly aims at pointing out the interest of a *functional* view of Boolean expressions when discussing analogical proportion-based inference. The paper first restates the notion of analogical proportion and its Boolean logic modeling. It then discusses the nature of the analogical proportion-based inference, first showing that it is a particular instance of a general "analogical jump" pattern, then explaining its link with affine Boolean functions, before discussing how we might take advantage of this situation for a better focusing of the analogical proportion-based inference.

2 Analogical proportion

Analogical proportions are statements of the form "a is to b as c is to d" like "Queen is to King as Woman is to Man", or "Paris is to France as Madrid is to Spain". In this paper, we assume that i) a, b, c, d are described in terms of Boolean features and they can be represented by the vector of their values on these features, and that ii) the relevant features are the same for a, b, c, and d. This second hypothesis is obviously not satisfied in the second example, indeed a and c belong to a conceptual universe (the one of cities) distinct from the one of b and d (the one of countries). Such more tricky proportions are discussed in [14].

2.1 Boolean logic modeling

The analogical proportion "a is to b as c is to d", denoted $a : b :: c : d$ in the following, intuitively suggests that a differs from b as c differs from d and b differs from a as d differs from c". In this subsection, a, b, c, and d are just Boolean variables, which pertain to the same unique feature for four items. The analogical proportion is logically expressed as [16] by the quaternary connective:

$$a : b :: c : d \triangleq ((a \wedge \neg b) \equiv (c \wedge \neg d)) \wedge ((\neg a \wedge b) \equiv (\neg c \wedge d)) \quad (1)$$

Note that this logical expression of an analogical proportion put forward dissimilarity, in agreement with the idea that analogy is as much a matter of dissimilarity as a matter of similarity. Similarity appears in the logically equivalent expression

$$a : b :: c : d = ((a \wedge d) \equiv (b \wedge c)) \wedge ((\neg a \wedge \neg d) \equiv (\neg b \wedge \neg c)) \quad (2)$$

This latter expression states that what a and d have in common (positively or negatively), b and c have it also in common.

Table 1 gives the truth table of $a : b :: c : d$. We can see that $a : b :: c : d$ is true for 6 patterns: 0000, 1111, 0011, 1100, 0101 and 1010 (in bold in Table 1) .

It is easy to see that the logical expression of $a : b :: c : d$ satisfies the key properties of an analogical proportion, namely

$a\ b\ c\ d$	$a : b :: c : d$	$a\ b\ c\ d$	$a : b :: c : d$
0 0 0 0	**1**	1 0 0 0	0
0 0 0 1	0	1 0 0 1	0
0 0 1 0	0	1 0 1 0	**1**
0 0 1 1	**1**	1 0 1 1	0
0 1 0 0	0	1 1 0 0	**1**
0 1 0 1	**1**	1 1 0 1	0
0 1 1 0	0	1 1 1 0	0
0 1 1 1	0	1 1 1 1	**1**

Table 1. Boolean valuations for $a : b :: c : d$

- reflexivity: $a : b : a : b$
- symmetry: $a : b :: c : d \Rightarrow c : d :: a : b$
- central permutation: $a : b :: c : d \Rightarrow a : c :: b : d$

Moreover, it is also worth noticing that the analogical proportion is independent with respect to the positive or negative encoding of a considered feature: $a : b :: c : d = \neg a : \neg b :: \neg c : \neg d$. Besides, with this definition, the analogical proportion is transitive in the following sense: $(a : b :: c : d) \wedge (c : d :: e : f) \Rightarrow a : b :: e : f$.

A simple extension of the definition of analogical proportion to Boolean vectors in $\mathbb{B}^n$ of the form $\boldsymbol{a} = (a_1, ..., a_n)$ is as follows: $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}$ iff $\forall i \in [1, n],\ a_i : b_i :: c_i : d_i$.

2.2 Equation solving

It is an acknowledged property of analogy to be creative. In this modeling, this is related to the following equation solving problem: find x such as $a : b :: c : x$ holds true, a, b, and c being given. It is easy to see that the equation has no solution in two cases: $1 : 0 :: 0 : x$ and $0 : 1 :: 1 : x$. When it exists the solution is clearly unique. It was first suggested by [11,12] that x can be computed as

$$x \triangleq c \equiv (a \equiv b)$$

where $\equiv$ is the equivalence connective $s \equiv t \triangleq (\neg s \vee t) \wedge (\neg t \vee s)$. Moreover, note that $s \equiv t = \neg((s \wedge \neg t) \vee (\neg s \wedge t)) = \neg(s \oplus t)$ where $\oplus$ is the xor connective (exclusive or). Thus it is clear that $c \equiv (a \equiv b)$ can be rewritten as $c \equiv (a \equiv b) = \neg(c \oplus \neg(a \oplus b)) = a \oplus b \oplus c$ since $\neg s = s \oplus 1$ and $1 \oplus 1 = 0$. Connectives $\equiv$ and $\oplus$ are associative operators. Thus, we can write $x = a \oplus b \oplus c$ and Table 2 shows the values of x in the 6 cases where equation $a : b :: c : x$ has a solution, as well as in the two remaining cases where there is no analogical solution for $a : b :: c : x$. In these two latter cases corresponding to patterns 0110 and 1001, we have a reverse analogy [19,20], where "b is to a as c is to d" holds rather than "a is to b as c is to d".

$\boldsymbol{a}$	0	1	0	1	0	1	0	1
$\boldsymbol{b}$	0	1	0	1	1	0	1	0
$\boldsymbol{c}$	0	1	1	0	0	1	1	0
$\boldsymbol{x}$	**0**	**1**	**1**	**0**	**1**	**0**	0	1

Table 2. Solving $a : b :: c : x$

Remark Interestingly enough, the eight patterns appearing in Table 2 with an *even* number of 0 and of 1 are involved in the four homogeneous logical proportions (which includes the analogical proportion and the reverse analogical proportion) [19]. The eight remaining patterns among the $2^4 = 16$ patterns of Table 1, which have an *odd* number of

0 and of 1 (and are at the basis of heterogeneous logical proportions [20]), appear in the columns of Table 3. The computation of the fourth line in such a case from the first three lines, in an equation now denoted $a/b//c/x$, is then given by $x = a \oplus b \oplus c \oplus 1$. This is an interesting operator that takes the majority value in a, b, c (the 6 first columns), provided that it does not lead to unanimity (the last two columns).

$\boldsymbol{a}$	1	0	0	1	0	1	0	1
$\boldsymbol{b}$	0	1	1	0	0	1	0	1
$\boldsymbol{c}$	0	1	0	1	1	0	0	1
$\boldsymbol{x}$	0	1	0	1	0	1	1	0

Table 3. Solving $a/b//c/x$

2.3 Analogical proportions induced by the comparison of two objects

As soon as we have two *distinct* Boolean vectors $\boldsymbol{a}$ and $\boldsymbol{d}$, it is possible to find two other vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ such that $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}$. Indeed, let $Agr(\boldsymbol{a}, \boldsymbol{d})$ be the set of indices where $\boldsymbol{a}$ and $\boldsymbol{d}$ agree and $Dis(\boldsymbol{a}, \boldsymbol{d})$ the set of indices where the two vectors differ. Let us now consider two new vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ such that : $\forall i \in Agr(\boldsymbol{a}, \boldsymbol{d}), a_i = b_i = c_i = d_i$ (all equal to 1 or all equal to 0) and $\forall i \in Dis(\boldsymbol{a}, \boldsymbol{d})(b_i = a_i$ and $c_i = d_i)$ or $(b_i = \neg a_i$ and $c_i = \neg d_i)$.

For instance, $\boldsymbol{a} = 0110, \boldsymbol{d} = 0011$, $Agr(\boldsymbol{a}, \boldsymbol{d}) = \{1, 3\}$ and $Dis(\boldsymbol{a}, \boldsymbol{d}) = \{2, 4\}$. Then $\boldsymbol{b} = 0111$ and $\boldsymbol{c} = 0010$ make $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}$ true. This may be viewed as instances of the equation solving problem $a : x :: x' : d$ with two unknowns x and x'. Obviously, we have always a solution: $x = a$ and $x' = d$ or $x = d$ and $x' = a$. But as soon as $Dif(\boldsymbol{a}, \boldsymbol{d})$ contains at least two indices as in the above example, we have solutions where the four vectors $\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{d}$ are distinct, as shown in the example. The creation of $(\boldsymbol{b}, \boldsymbol{c})$ from $\boldsymbol{a}$ and $\boldsymbol{d}$ is illustrated in [10] on images, using a non logical approach.

2.4 Non Boolean attributes

Real life datasets rarely involve Boolean features only. There may be a mix between Boolean and nominal feature (like $color$), or real-valued features. In the case of nominal attributes, it is quite common to binarise in the following way: for instance $color$ can take three values $red, green, blue$ which will be coded as $100, 010, 001$ (using features as $isRed, isGreen, isBlue$). Then we are back to the Boolean case. The case of real-valued features is more sophisticated and needs the tool of multi-valued logic to be properly handled. We refer the interested reader to [21,7] for a comprehensive development. Nevertheless, in this paper, we strictly stick to the Boolean case.

Another important issue is to get the relevant feature to code a given problem. We do not focus on this issue here as we consider the vectors coming from existing datasets, so the coding has already been done.

3 Analogical proportion-based inference

We have seen that we can obtain the solution x, when it exists, of an analogical proportion equation $a : b :: c : x$ as $x = a \equiv b \equiv c = a \oplus b \oplus c$. The analogical proportion-based inference principle [23] can now be stated as follows:

$$\frac{\forall i \in \{1, ..., n\},\ \ a_i : b_i :: c_i : d_i \text{ holds}}{\forall j \in \{n+1, ..., p\},\ \ a_j : b_j :: c_j : d_j \text{ holds}}$$

This is a form of analogical reasoning where we transfer knowledge from some components of our vectors to their remaining components, tacitly assuming that the values of the n first components determine the values of the others. An important particular case of this pattern is when $p = n + 1$, which corresponds to the situation in classification where the $(n + 1)$th component corresponds to the class of the item described by the n first features. Note also that this pattern is a tool for guessing missing values in a table, a problem, which has been considered for a long time [1]. Let us now examine how this inference pattern can be related to a more usual "analogical jump" pattern.

3.1 An instance of a general "analogical jump" pattern

In its simplest form, analogical reasoning, without any reference to the notion of proportion, is usually viewed as a way to infer some new fact on the basis of a single observation. Analogical reasoning has been mainly formalized in the setting of first order logic [6,13], and in second order logic [9]. A basic pattern for analogical reasoning is then to consider 2 terms s and t, to observe that they share a property P, and knowing that another property Q also holds for s, to infer that it holds for t as well. This is known as the "analogical jump" and can be described with the following simplified inference pattern, leading (possibly) to a wrong conclusion:

$$\frac{P(s)\ \ P(t)\ \ Q(s)}{Q(t)} \quad (AJ)$$

Making such an inference pattern valid would require the implicit hypothesis that P determines Q inasmuch as $\nexists u\ P(u) \wedge \neg Q(u)$. This may be ensured if there exists an underlying functional dependency, or more generally, if it is known for instance that when something is true for an object of a certain type, then it is true for all objects of that type. Otherwise, without such guarantees, the result of an analogical inference may turn to be definitely wrong.

To link the above analogical pattern with the concept of analogical proportion, it is tempting to write something like: $P(s) : P(t) :: Q(s) : Q(t)$ since we have 4 terms which obey, at least from a syntactic viewpoint, the structure of an analogical proportion. Indeed, it is sufficient to encode each piece of information in a binary way according to the presence or the absence of P, Q, s, or t in the corresponding term, and we get the encoding d of $Q(t)$ via the equation solving process as in Table 4. In that

	P	Q	s	t	
$\boldsymbol{a}$	1	0	1	0	$P(s)$
$\boldsymbol{b}$	1	0	0	1	$P(t)$
$\boldsymbol{c}$	0	1	1	0	$Q(s)$
$\boldsymbol{d}$	0	1	0	1	$Q(t)$

Table 4. A syntactic view of analogical jump

case, $\boldsymbol{a} = P(s), \boldsymbol{b} = P(t), \boldsymbol{c} = Q(s), \boldsymbol{d} = Q(t)$ are encoded as Boolean vectors where the semantics carried by the predicate symbols P and Q is not considered.

In [26,2,15,3], the authors take a similar inspiration where, starting from Boolean datasets and focusing on binary classification problem, they apply the following inference principle (and obtain competitive results on benchmark data sets):

$$\frac{\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}}{cl(\boldsymbol{a}) : cl(\boldsymbol{b}) :: cl(\boldsymbol{c}) : cl(\boldsymbol{d})} \quad AP$$

It means that if 4 Boolean vectors build a valid analogical proportion, then it should be true that their classes build also a valid proportion. Starting from this viewpoint, in the case where $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ are in a sample set, i.e., their classes are known, and $\boldsymbol{d}$ being the object to be classified, if the equation $cl(\boldsymbol{a}) : cl(\boldsymbol{b}) :: cl(\boldsymbol{c}) : x = 1$ is solvable (in that case, we say that the triple $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ is *class solvable*), they allocate its solution to $cl(\boldsymbol{d})$ just by applying the previous principle. Experiments highlight the predictive power of this principle. Let us understand why AP principle is just a particular instance of (AJ):

- Considering $\boldsymbol{a}$ and $\boldsymbol{b}$ as Boolean vectors in $\mathbb{B}^n$, the vector $\boldsymbol{k} = \boldsymbol{a} - \boldsymbol{b}$ belongs to $\{-1, 0, 1\}^n$ and summarizes the result of the comparison between $\boldsymbol{a}$ and $\boldsymbol{b}$. So given such a vector k, we define the predicate $P_k(\boldsymbol{a}, \boldsymbol{b}) := (\boldsymbol{a} - \boldsymbol{b} = \boldsymbol{k})$.
- Then we can consider 3 predicate symbols Q_1, Q_2, Q_3 defined as follows:
  1. $Q_1(\boldsymbol{a}, \boldsymbol{b}) := (cl(\boldsymbol{a}) = cl(\boldsymbol{b}))$
  2. $Q_2(\boldsymbol{a}, \boldsymbol{b}) := (cl(\boldsymbol{a}) = 0) \wedge (cl(\boldsymbol{b}) = 1))$
  3. $Q_3(\boldsymbol{a}, \boldsymbol{b}) := (cl(\boldsymbol{a}) = 1) \wedge (cl(\boldsymbol{b}) = 0))$

Let us note that the Q_i's are pairwise mutually exclusive predicates. Using these predicate symbols, the following rule:

$$\frac{P_k(\boldsymbol{a}, \boldsymbol{b}) \;\; P_k(\boldsymbol{c}, \boldsymbol{d}) \;\; Q_i(\boldsymbol{a}, \boldsymbol{b})}{Q_i(\boldsymbol{c}, \boldsymbol{d})}$$

is just an instance of (AJ). Moreover, it states that when the difference $\boldsymbol{a} - \boldsymbol{b}$ equals to $\boldsymbol{c} - \boldsymbol{d}$, then the relation between $cl(\boldsymbol{a})$ and $cl(\boldsymbol{b})$ is the same as the relation between $cl(\boldsymbol{c})$ and $cl(\boldsymbol{d})$. If we notice that $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}$ is just equivalent to $P_k(\boldsymbol{a}, \boldsymbol{b}) \wedge P_k(\boldsymbol{c}, \boldsymbol{d})$ (in the exact sense of the formal definition of the analogical proportion applied componentwise), and $Q_i(\boldsymbol{a}, \boldsymbol{b}) \wedge Q_i(\boldsymbol{c}, \boldsymbol{d})$ entails $cl(\boldsymbol{a}) : cl(\boldsymbol{b}) :: cl(\boldsymbol{c}) : cl(\boldsymbol{d})$, we obtain:

$$\frac{\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}}{cl(\boldsymbol{a}) : cl(\boldsymbol{b}) :: cl(\boldsymbol{c}) : cl(\boldsymbol{d})}$$

which is exactly the expression of AP used in [2].

In pattern AP, we transfer the identity of differences pertaining to pairs $(\boldsymbol{a}, \boldsymbol{b})$ and $(\boldsymbol{c}, \boldsymbol{d})$ to the relation between their classes. It enables us to predict the missing information about $\boldsymbol{d}$, using AP as an extrapolation principle. This is clearly a form of reasoning that is not sound, but which may be useful for trying to guess unknown values.

3.2 Link with affine Boolean functions

As already mentioned [4], an analogical proportion of the form "$cl(\boldsymbol{b})$ is to $cl(\boldsymbol{a})$ as $\boldsymbol{b}$ is to $\boldsymbol{a}$" sounds a bit like the expression of the *qualitative derivative* of a function cl underlying the classification process, since the derivative of a function f in $\boldsymbol{a}$ is the limit when $x \longrightarrow a$ of the ratio $\frac{f(\boldsymbol{x})-f(\boldsymbol{a})}{\boldsymbol{x}-\boldsymbol{a}}$, which is a matter of comparing two algebraic differences[1]. Moreover, considering the 6 patterns that make true the analogical proportion, it can be also noticed that if there is a change from $\boldsymbol{a}$ to $\boldsymbol{b}$, there should a change in the same direction from $\boldsymbol{c}$ to $\boldsymbol{d}$. Besides, it has been observed that once extended from Boolean to graded scales, the analogical proportion-based inference provides a *linear* interpolation mechanism [7]. This is due to the fact that in this case $a : b :: c : d = 1$ if and only if $d - c = b - a$ (where a, b, c, $d \in [0, 1]$) and the solution of the equation $a : x :: x : d = 1$ is $x = \frac{a+d}{2}$.

In fact, it has been recently formally proved [5] that the AP principle is sound as soon as the labeling function is an affine Boolean function (this means in practice that the function is a constant, a projection, an xor function, or an $\equiv$ function, over some subsets of the n attributes). Moreover it can be shown that there is no other Boolean function for which this is true [5].

It is well-known that any Boolean function can be put in a polynomial form where the sum is taken as $\oplus$ and the product as the $\min$ [22]. With a functional view of Boolean expressions in mind, one can use the notion of qualitative derivative and define a Taylor-like development of a Boolean function; see, e.g., [17,25]. For instance, consider the linear function $f(x, y) = x \oplus y$ (indeed a polynomial of degree 1 and arity 2). Then it can be checked (see Table 5), that we can write a Taylor-like development of the form:

$$f(x, y) = f(a, b) \oplus \partial^f_x(a) \wedge (x \ominus a) \oplus \partial^f_y(b) \wedge (y \ominus b) \triangleq \Sigma(x, y)$$

where $s \ominus t = s \oplus t$ since $s \ominus t = x \Leftrightarrow s = t \oplus x$, and where all the partial derivatives are equal to 1 here. One can also rewrite the above equality as $f(x, y) \ominus f(a, b) = (x \ominus a) \oplus (y \ominus b)$, which is indeed the Boolean counterpart of what holds for affine functions in $\mathbb{R}^n$.

$x\ y\ a\ b$	$f(x,y) = x \oplus y$	$f(a,b) = a \oplus b$	$\partial^f_x(a) \wedge (x \ominus a)$	$\partial^f_y(b) \wedge (y \ominus b)$	$\Sigma(x,y)$
0 0 0 0	0	0	$1 \wedge (0 \ominus 0) = 0$	$1 \wedge (0 \ominus 0) = 0$	0
0 0 0 1	0	1	$1 \wedge (0 \ominus 0) = 0$	$1 \wedge (0 \ominus 1) = 1$	0
0 0 1 0	0	1	$1 \wedge (0 \ominus 1) = 1$	$1 \wedge (0 \ominus 0) = 0$	0
0 0 1 1	0	0	$1 \wedge (0 \ominus 1) = 1$	$1 \wedge (0 \ominus 1) = 1$	0
0 1 0 0	1	0	$1 \wedge (0 \ominus 0) = 0$	$1 \wedge (1 \ominus 0) = 1$	1
0 1 0 1	1	1	$1 \wedge (0 \ominus 0) = 0$	$1 \wedge (1 \ominus 1) = 0$	1
0 1 1 0	1	1	$1 \wedge (0 \ominus 1) = 1$	$1 \wedge (1 \ominus 0) = 1$	1
0 1 1 1	1	0	$1 \wedge (0 \ominus 1) = 1$	$1 \wedge (1 \ominus 1) = 0$	1
1 0 0 0	1	0	$1 \wedge (1 \ominus 0) = 1$	$1 \wedge (0 \ominus 0) = 0$	1
1 0 0 1	1	1	$1 \wedge (1 \ominus 0) = 1$	$1 \wedge (0 \ominus 1) = 1$	1
1 0 1 0	1	1	$1 \wedge (1 \ominus 1) = 0$	$1 \wedge (0 \ominus 0) = 0$	1
1 0 1 1	1	0	$1 \wedge (1 \ominus 1) = 0$	$1 \wedge (0 \ominus 1) = 1$	1
1 1 0 0	0	0	$1 \wedge (1 \ominus 0) = 1$	$1 \wedge (1 \ominus 0) = 1$	0
1 1 0 1	0	1	$1 \wedge (1 \ominus 0) = 1$	$1 \wedge (1 \ominus 1) = 0$	0
1 1 1 0	0	1	$1 \wedge (1 \ominus 1) = 0$	$1 \wedge (1 \ominus 0) = 1$	0
1 1 1 1	0	0	$1 \wedge (1 \ominus 1) = 0$	$1 \wedge (1 \ominus 1) = 0$	0

Table 5. Taylor-like expression of the linear function $f(x, y) = x \oplus y$

[1]We may also remember that the idea of differential has inspired adaption methods in case-based reasoning for solving numerical problems [8], even if case-based reasoning deals with cases one by one rather than handling triples of cases.

3.3 Adjusting analogical proportion-based inference

Since analogical proportion-based inference works perfectly for predicting affine Boolean functions, it is natural to wonder if given a training set $\mathcal{T}$ it would not be possible to cover it with a *piecewise* affine Boolean function.

The answer is yes, ... and in many different ways! It is easy to see that we just need two pieces. Indeed let $f(x_1, \cdots, x_n)$ be an affine function. In a binary classification problem, any function (be affine or not) partitions $\mathcal{T}$ into two parts $\mathcal{T}_f$ and $\mathcal{T}_{f\oplus 1}$ on which respectively f and $f \oplus 1$ correctly predict the class (since $f \oplus 1$ is just $\neg f$).

If the classification in $\mathcal{T}$ obeys an affine Boolean function f then either $\mathcal{T}_f$ or $\mathcal{T}_{f\oplus 1}$ is empty. This also means that the application of the analogical-proportion principle will amount to apply function f to a new item to be classified (or function $f \oplus 1$ if $\mathcal{T}_f$ is empty). So if the training set $\mathcal{T}$ is covered by an "almost" linear function, this means that one of the two subsets of the partition of $\mathcal{T}$ is very large with respect to the other.

When the training set is not covered by a unique affine Boolean function, this means that there exist triples that lead to false predictions when applying AP. Then we might think that it happens more often when a, b, c do not all belong to the same subset in some partition induced by a function f. So an idea for "almost" linear functions, would be to look for the affine Boolean functions such as $\mathcal{T}_f$ (or $\mathcal{T}_{f\oplus 1}$) is the largest possible subset, which would make easier the finding of triples such as all the three $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ are in it. However another issue is to wonder if some partitions are more appropriate than others for guessing the class of a particular new item.

Generally speaking, the issue is to find a way to identify those triples that are "suspect", i.e., likely to yield a faulty prediction, among a set of candidate triples that enables you to apply AP. Indeed in case of multiple triples, which is the usual situation, we apply a voting procedure among the predictions of the applicable triples, where sometimes the faulty triples are the majority. How to restrict this voting procedure to "good triples"? Another idea may come from a careful examination of the way triples are built and of the meaning of pairs inside, as first suggested in [4].

In Table 6, we have reordered the vectors in a particular way. Indeed the table shows that building the analogical proportion $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}$ is a matter of pairing the pair $(\boldsymbol{a}, \boldsymbol{b})$ with the pair $(\boldsymbol{c}, \boldsymbol{d})$. More precisely, on features or attributes $\mathcal{A}_1$ to $\mathcal{A}_{j-1}$, the four vectors are equal; on attributes $\mathcal{A}_j$ to $\mathcal{A}_{r-1}$, $\boldsymbol{a} = \boldsymbol{b}$ and $\boldsymbol{c} = \boldsymbol{d}$, but $(\boldsymbol{a}, \boldsymbol{b}) \neq (\boldsymbol{c}, \boldsymbol{d})$. In other words, on attributes $\mathcal{A}_1$ to $\mathcal{A}_{r-1}$ $\boldsymbol{a}$ and $\boldsymbol{b}$ agree and $\boldsymbol{c}$ and $\boldsymbol{d}$ agree as well. This contrasts with attributes $\mathcal{A}_r$ to $\mathcal{A}_n$, for which we can see that $\boldsymbol{a}$ differs from $\boldsymbol{b}$ as $\boldsymbol{c}$ differs from $\boldsymbol{d}$ (and vice-versa). In columns we recognize the 6 patterns that makes the analogical proportion true. There are two cases, either $cl(\boldsymbol{a}) = cl(\boldsymbol{b})$ (and then $cl(\boldsymbol{x}) = cl(\boldsymbol{c})$), or $cl(\boldsymbol{a}) \neq cl(\boldsymbol{b})$ (and then $cl(\boldsymbol{x}) = cl(\boldsymbol{b})$). In the first case, it suggests that the particular change observed between $\boldsymbol{a}$ and $\boldsymbol{b}$ on features from $\mathcal{A}_r$ to $\mathcal{A}_n$ does not affect cl in the context defined by the values of the features from $\mathcal{A}_1$ to $\mathcal{A}_{r-1}$ where $\boldsymbol{a}$ and $\boldsymbol{b}$ are equal. Applying AP amounts to assuming that this absence of effect is true in other contexts of values of features from $\mathcal{A}_1$ to $\mathcal{A}_{r-1}$. So the smaller the number of features from $\mathcal{A}_r$ to $\mathcal{A}_n$, the more cautious. A similar reasoning can be done when $cl(\boldsymbol{a}) \neq cl(\boldsymbol{b})$ where the change on the features from $\mathcal{A}_r$ to $\mathcal{A}_n$ should be responsible of the change of class in the context of the values of the other attributes. Observe also that if we have two pairs $(\boldsymbol{a}, \boldsymbol{b})$ and $(\boldsymbol{a'}, \boldsymbol{b'})$ such as $\boldsymbol{a'} : \boldsymbol{b'} :: \boldsymbol{a} : \boldsymbol{b}$, while $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{x}$, then by transitivity we have $\boldsymbol{a'} : \boldsymbol{b'} :: \boldsymbol{c} : \boldsymbol{x}$. Thus transitivity agrees with the idea that if a change has an effect (or no effect) in some context, then it may be the same elsewhere.

	$\mathcal{A}_1$	...	$\mathcal{A}_{i-1}$	$\mathcal{A}_i$	...	$\mathcal{A}_{j-1}$	$\mathcal{A}_j$	...	$\mathcal{A}_{k-1}$	$\mathcal{A}_k$	...	$\mathcal{A}_{r-1}$	$\mathcal{A}_r$	...	$\mathcal{A}_{s-1}$	$\mathcal{A}_s$	...	$\mathcal{A}_n$	cl
$\boldsymbol{a}$	1	...	1	0	...	0	1	...	1	0	...	0	1	...	1	0	...	0	$cl(\boldsymbol{a})$
$\boldsymbol{b}$	1	...	1	0	...	0	1	...	1	0	...	0	0	...	0	1	...	1	$cl(\boldsymbol{b})$
$\boldsymbol{c}$	1	...	1	0	...	0	0	...	0	1	...	1	1	...	1	0	...	0	$cl(\boldsymbol{c})$
$\boldsymbol{x}$	1	...	1	0	...	0	0	...	0	1	...	1	0	...	0	1	...	1	$cl(\boldsymbol{x})?$

Table 6. Pairing pairs (a, b) and (c, d)

A last idea would be to consider "continuous" analogical proportion and to solve interpolative equation $\boldsymbol{a} : \boldsymbol{x} :: \boldsymbol{x} : \boldsymbol{b}$. In a Boolean setting, such an equation has no solution, except in the trivial situation where $\boldsymbol{a} = \boldsymbol{b}$, then $\boldsymbol{x} = \boldsymbol{a}$. In the associated class equation one has necessarily $cl(\boldsymbol{a}) = cl(\boldsymbol{x}) = cl(\boldsymbol{b})$. Then one may relax $\boldsymbol{a} : \boldsymbol{x} :: \boldsymbol{x} : \boldsymbol{b}$ to a subset of features and makes sure that $\boldsymbol{x}$ is between $\boldsymbol{a}$ and $\boldsymbol{b}$ in the sense that $\max(h(\boldsymbol{a}, \boldsymbol{x}), h(\boldsymbol{x}, \boldsymbol{b})) \leq h(\boldsymbol{a}, \boldsymbol{b})$, where h is the Hamming distance. In such a case, we have a variant of nearest neighbors methods.

We have emphasized the role played by affine Boolean functions, suggesting that the training set in a classification problem might be restricted to subsets of examples more relevant for a new item to be classified. These subsets of examples should be covered by some affine Boolean function. Finding them remains an open question.

4 Conclusion

The paper has intended to provide an advanced discussion of the analogical proportion-based inference principle in the Boolean case, in a classification perspective. As already said, analogical proportion-based inference is also available for nominal and real valued data. The application of analogical proportions to regression is an open problem; then the agreement between a qualitative and a quantitative view of these proportions is crucial (see, e.g., [24] on such issue in learning); in that respect the main gradual extension [21,7] clearly distinguishes between situations where the changes from $\boldsymbol{a}$ to $\boldsymbol{b}$ and from $\boldsymbol{c}$ to $\boldsymbol{d}$ are in the same direction, and where the changes are in opposite directions.

Generally speaking, some authors, e.g., [18], view qualitative reasoning as made of components such as: comparison, categorization, identification of relations, and emergence of a meaning. Analogical proportions seem to offer an interesting mixture of at least two or three of these ingredients [14]; the proper understanding of their interrelationships is still to be further explored.

References

1. M. Abraham, D. M. Gabbay, and U. J. Schild. Analysis of the talmudic argumentum a fortiori inference rule (kal vachomer) using matrix abduction. *Studia Logica*, 92(3):281–364, 2009.
2. S. Bayoudh, L. Miclet, and A. Delhay. Learning by analogy: A classification rule for binary and nominal data. *Proc. Int. Joint Conf. on Artif. Intellig. (IJCAI'07)*, pages 678–683, 2007.
3. M. Bounhas, H. Prade, and G. Richard. Analogical classification: A new way to deal with examples. In *Proc. 21st Eur.Conf. Artif. Intell. (ECAI'14), Prague*, IOS Press,135-140, 2014.
4. W. Correa, H. Prade, and G. Richard. Trying to understand how analogical classifiers work. In E. Hüllermeier et al., editor, *Proc. 6th Inter. Conf. on Scalable Uncertainty Management (SUM'12), Marburg, Germany*, LNAI 7520, pages 582–589. Springer Verlag, 2012.

5. M. Couceiro, N. Hug, H. Prade, and G. Richard. Analogy-preserving functions: A way to ext-end Boolean samples. In *Proc.26th Int. J. Conf. Artif. Intel. IJCAI'17, Melbourne*, 2017.
6. T. R. Davies and Russell S. J. A logical approach to reasoning by analogy. In J. P. McDermott, editor, *Proc. of the 10th International Joint Conference on Artificial Intelligence (IJCAI'87), Milan*, pages 264–270. Morgan Kaufmann, 1987.
7. D. Dubois, H. Prade, and G. Richard. Multiple-valued extensions of analogical proportions. *Fuzzy Sets and Systems*, 292:193–202, 2016.
8. B. Fuchs, J. Lieber, A. Mille, and A. Napoli.Differential adaptation: An operational approach to adaptation for solving numerical problems with CBR. *Know.-Bd. Sys.*, 68:103–114, 2014.
9. H. Gust, K. Kühnberger, and U. Schmid. Metaphors and heuristic-driven theory projection (HDTP). *Theoretical Computer Science*, 354(1):98 – 117, 2006.
10. L. Jing, Y. Lu, H. Gang, and K. Sing Bing. Visual attribute transfer through deep image analogy. In *ACM Transactions on Graphics, (Proc. of Siggraph)*, volume 36, 2017.
11. S. Klein. Culture, mysticism & social structure and the calculation of behavior. In *Proc. 5th Europ. Conf. in Artificial Intelligence (ECAI'82), Orsay, France*, pages 141–146, 1982.
12. S. Klein. Analogy and mysticism and the structure of culture (and Comments & Reply). *Current Anthropology*, 24 (2):151–180, 1983.
13. E. Melis and M. Veloso. Analogy in problem solving. In *Handbook of Practical Reasoning: Computational and Theoretical Aspects*. Oxford Univ. Press, 1998.
14. L. Miclet, N. Barbot, and H. Prade. From analogical proportions in lattices to proportional analogies in formal concepts. In T. Schaub, G. Friedrich, and B. O'Sullivan, editors, *Proc. 21st Europ. Conf. on Artificial Intelligence (ECAI'14), Prague, Aug. 18-22*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 627–632. IOS Press, 2014.
15. L. Miclet, S. Bayoudh, and A. Delhay. Analogical dissimilarity: Definition, algorithms and two experiments in machine learning. *J. Artif. Intell. Res. (JAIR)*, 32:793–824, 2008.
16. L. Miclet and H. Prade. Handling analogical proportions in classical logic and fuzzy logics settings. In *Proc. 10th Eur. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'09),Verona*, pages 638–650. Springer, LNCS 5590, 2009.
17. E. Mizraji. Differential and integral calculus for logical operations. a matrix-vector approach. *J. Log. Comput.*, 25(3):613–638, 2015.
18. A. Mucchielli. Les processus intellectuels fondamentaux sous-jacents aux techniques et mthodes qualitatives. *RecherchesQualitatives*, 3:1–27, 2006.
19. H. Prade and G. Richard. From analogical proportion to logical proportions. *Logica Universalis*, 7(4):441–505, 2013.
20. H. Prade and G. Richard. Homogenous and heterogeneous logical proportions. *IfCoLog J. of Logics and their Applications*, 1(1):1–51, 2014.
21. H. Prade and G. Richard. Multiple-valued logic interpretations of analogical, reverse analogical, and paralogical proportions. In *Proc. 40th IEEE Int. Symp. on Multiple-Valued Logic (ISMVL'10)*, pages 258–263, Barcelona, 2010.
22. M. H. Stone. The theory of representation for Boolean algebras. *Trans. of the American Mathematical Society*, 40(1):37111, 1936.
23. N. Stroppa and F. Yvon. Analogical learning and formal proportions: Definitions and methodological issues. Technical Report D004, ENST-Paris, 2005.
24. Dorian Suc, Daniel Vladusic, and Ivan Bratko. Qualitatively faithful quantitative prediction. *Artificial Intelligence*, 158(2):189 – 214, 2004.
25. S. N. Yanushkevich and V. P. Shmerko. Taylor expansion of logic functions: From conventional to nanoscale design. Int. TICSP Wksp Spectral Meth. & Multirate Signal Process.,2004.
26. F. Yvon and N. Stroppa. Formal models of analogical proportions. Technical Report D008, Ecole Nationale Supérieure des Télécommunications, Paris, France, 2006.

A Convolutional Siamese Network for Developing Similarity Knowledge in the SelfBACK Dataset

Kyle Martin, Nirmalie Wiratunga, Sadiq Sani, Stewart Massie, Jeremie Clos

School of Computing Science and Digital Media,
Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
{ k.martin | n.wiratunga | s.sani | s.massie | j.clos }@rgu.ac.uk

Abstract: The Siamese Neural Network (SNN) is a neural network architecture capable of learning similarity knowledge between cases in a case base by receiving pairs of cases and analysing the differences between their features to map them to a multi-dimensional feature space. This paper demonstrates the development of a Convolutional Siamese Network (CSN) for the purpose of case similarity knowledge generation on the SelfBACK dataset. We also demonstrate a CSN is capable of performing classification on the SelfBACK dataset to an accuracy which is comparable with a standard Convolutional Neural Network.

Keywords: Case-Based Reasoning · Siamese Neural Networks · Categorisation · SelfBACK

1 Introduction

Similarity knowledge is an essential component of an effective Case-Based Reasoning (CBR) system, but its generation can be a daunting task. Large complex datasets, where inter-feature relationships may exist, present a challenge to traditional similarity generation measures. Although similarity-based retrieval can offer numerous advantages during the retrieval phase, this can have a large initial cost. It is little wonder that recent research is targeting methods of harnessing deep learning methods to improve similarity knowledge generation.

A Siamese Neural Network (SNN) is a deep learning architecture which can learn similarity knowledge at a case-to-case level. SNNs have proven effective at learning similarity knowledge for a range of different domains including smartphone gesture classification and face verification [2, 4]. This paper presents the application of an SNN architecture to the SelfBACK[1] dataset[2], which contains

[1]The SelfBACK project is funded by European Union's H2020 research and innovation programme under grant agreement No. 689043. More details available: http://www.selfback.eu

[2]The SelfBACK dataset associated with this paper is publicly accessible from https://github.com/selfback/activity-recognition

the accelerometer data for 34 users labelled with one of 6 activities. The main contribution of this paper is to demonstrate the successful application of an SNN as a means to develop similarity knowledge within a case base. In addition, this paper demonstrates that an SNN can perform a classification task on a level which is competitive with a typical Convolutional Neural Network (CNN).

This paper is organised into the following sections. Section 2 gives an overview of the research regarding learning similarity measure for use in CBR, as well as the SNN architecture and how it may be used as a means to develop similarity knowledge between cases which can be used for classification. Section 3 contains a description of the SelfBACK dataset and how it was used within the context of the presented research. Section 3 also describes our evaluation and details the setup of our experiments, including our classification method and network architecture, as well as pair creation method. Section 4 contains the results of our experiments and Section 5 highlights further work we aim to complete within this research area.

2 Related Works

2.1 Learning Similarity Measures

Learning effective similarity measures between cases can counter many of the issues that plague the retrieval phase of CBR systems, such as retrieving suitable results from extremely large and complex case bases, or retrieving results for cases where some features cannot be explicitly described [10]. However, the process of learning similarity knowledge can itself present an issue, and as such it has been the focus of much research.

Knowledge-Intensive Similarity Measures (KISMs) have been shown to improve retrieval in case bases where domain-specific knowledge is a key component [11]. While the main intention of standard similarity measures is to numerically quantify the simillitude between two cases based upon explicit feature values, K-ISMs use domain-specific knowledge to weight more important features for return [5]. This has been shown to improve retrieval accuracy in complex domains, and domains that rely on expert knowledge to query. However, the acqusition and encoding of domain-knowledge into similarity measures is an extremely expensive process which can often require the input of a domain expert. One of the advantages of the presented SNN architecture is that it can weight features automatically without the input of a domain expert and is significantly less time-consuming.

2.2 The Siamese Neural Network Architecture

An SNN architecture consists of two neural networks that share identical weights and are joined at one or more layers. SNNs receive case pairs as input to both the training and testing phases to develop simlarity knowledge at an object-to-object level. An example architecture is shown in Figure 1. During the training phase, these pairs are labelled as either 'genuine' (if the examples share the same class) or 'impostor' (if the examples are of different classes). This allows the network to develop a multi-dimensional space based upon cases features, where 'genuine' pairs are pushed closer together and 'impostor' pairs are pulled further away from each other. The output of the identical neural networks (or 'sub-networks') are feature vectors for each member of the input pair. The distance between these vectors is measured at the similarity layer to ascertain whether they belong to the same class based upon a threshold.

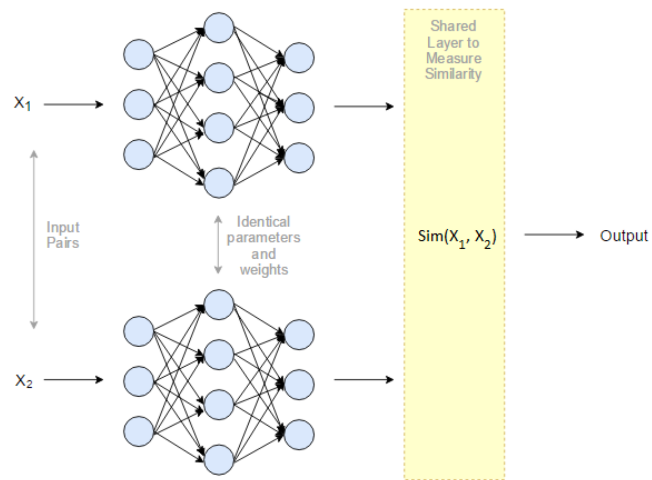


Figure 1: Siamese Neural Network Architecture

SNNs use 'contrastive loss', which was introduced in [4]. Contrastive loss is calculated by summing the results of the individual loss formulas for genuine and impostor pairs. Genuine pairs are penalized by loss L_G for being too far apart, while negative pairs are penalized by L_I if their distance falls within the given margin value. Sub-network weights are then updated by backpropagating the loss with respect to the weights. This means that genuine pairs are pushed closer together over the course of training, whilst ensuring that impostor pairs maintain at least a set distance apart. The similarity metric is therefore directly learned by the network, as it is implicitly defined by the loss function.

The equations for contrastive loss are detailed in Equations (1), (2) and (3). Y_A and Y_P are binary values which are equal to 0 for genuine pairs and 1 for impostor pairs, where Y_A is the actual label, Y_P is the predicted label and M is the margin.

$$L_G = (1 - Y_A){Y_P}^2 \tag{1}$$

$$L_I = Y_A(max(M - Y_P, 0))^2 \tag{2}$$

$$L = L_G + L_I \tag{3}$$

2.3 Similarity and Classification in SNNs

Initially made popular by [4] to identify similarities for face verification, many research efforts have taken advantage of an SNN's capability to develop similarity knowledge, in areas ranging from smartphone gesture classification [2], to similar text retrieval [11]. In [11], the authors demonstrate that their SNN can outperform state-of-the-art text similarity measures by mapping term vectors to a low dimensional space. Their results indicate that the SNN can significantly outperform other methods on both low and high dimensional data. The drawback was that the algorithm did not scale well to large amounts of examples.

Although introduced as a method of signature verification and binary classification [3], recent research has shown that SNNs are able to generalise to multiclass classification. In [6], the authors demonstrate that a Convolutional Siamese Network (CSN) can achieve very close to the state of the art and human levels of recognition in a one-shot learning setting on the omniglot dataset, which contains 40 distinct classes. A CSN is a type of SNN where the parallel neural networks are replaced with two identical Convolutional Neural Networks (CNNs). A CNN itself is a feed forward neural network which arranges its neurons in multiple dimensions in order to operate effectively on high dimensional data. One of the main advantages of using CNNs is that they can learn local feature detectors and are fairly robust to distortions of network input [7].

In [1], the authors demonstrated visual search on multiple domains by performing nearest neighbour on the output feature vectors from an CSN. Their findings showed that CSNs could be used to learn the similarity between images and that a nearest neighbour algorithm could be performed to retrieve the most similar images for a given example. Their findings demonstrated that the feature vectors produced by a CSN have potential use as a means to increase utility of more conventional classification techniques.

We can observe from the range of examples above that SNNs are capable of learning similarity knowledge and performing classification upon a wide range of domains. However, there remain areas which require further exploration. In particular, literature regarding the structuring of pair creation is lacking, as is research which utilises SNNs within ensemble classifiers.

3 Evaluation

The aims of this paper are two-fold; to show that an SNN could generate similarity knowledge within a case base, and to demonstrate the performance of an SNN as a method of human activity classification. To this end, we performed two experiments upon the SelfBACK dataset.

3.1 The Dataset

The SelfBack dataset consists of time series data collected from 34 users performing different activities over a short period of time. Data was collected by mounting a tri-axial accelerometer on the thigh and right-hand wrist of participants at a sampling rate of 100Hz as they completed a script of set activities, performing each for an average of three minutes [9]. Frequency coefficients were obtained by applying Discrete Cosine Transforms (DCT) and Discrete Fourier Transforms (FFT) to the raw accelerometer data.

Our experiments used the thigh dataset due to time limitations. Data was split into 5 second windows, meaning that there were between 160 and 180 cases per user and 1,500 features per case. This resulted in 6,084 cases of thigh data. These were then labelled as one of 6 activities (standing, upstairs, downstairs, walking, jogging, sitting) to create the full dataset.

3.2 Experimental Setup

Firstly, we implemented a Convolutional Siamese Network (CSN) upon the raw accelerometer thigh data, DCT thigh data and FFT thigh data. Pairs of cases were fed into the CSN and the convolutional sub-networks of the CSN learned to produce representative feature vectors of each case. We tested that the network had learned feature vectors which were representative of the original cases by measuring the euclidean distance between pair vectors at the similarity layer and comparing this to a threshold to identify whether a pair of examples belonged to the same class (a genuine pair) or to different classes (an impostor pair). If we identified that the distance between the genuine pair was less than a certain threshold (i.e. the space that should exist between cases of opposing classes) and the distance between the impostor pair was greater than this threshold, then we could reasonably assume that the case had been mapped to the correct space (or a very close approximation of it). We therefore used the percentage of correctly identified pairs as our accuracy metric.

Secondly, we implemented a CSN and CNN to perform classification on the raw thigh accelerometer data. The raw data was used in order to demonstrate a comparison between the two architectures which was unaffected by preprocessing of the data. For this experiment, we completed a similar process to the previous experiment until the CSN had learned representative feature vectors for each case in the test set. Each feature vector from the test set was then compared with 6 randomly selected class representative vectors generated from cases in the training set. The distance between the unlabelled test vector and each class representative vector was measured, and the test case was identified as belonging to the same class as the nearest class representative vector. The accuracy of the experiment was the percentage of correctly classified cases.

3.3 Network Architecture

A CSN was constructed from 2 sub-networks, which had 2 convolutional layers, a flattening layer and 2 fully connected layers. The first layers used tanh activation functions, while the final layer used a softmax function. The network was optimised using Stochastic Gradient Descent and the hyperparameters in Table 1. The output of the sub-networks was a representative feature vector for each case member of the pair. Euclidean distance between these vectors could then be measured and compared with the threshold.

Parameter	Value
Learning Rate	0.01
Learning Decay	0.000001
Nesterov Momentum	0.9

Table 1: CSN Hyperparameter Settings

In order to be comparable with the CSN, the CNN created for comparison purposes was a close replica of one of the sub-networks outlined above. The only major difference was the use of categorical cross-entropy for the loss function and backpropagating this loss with respect to the weights.

Implementations were run for10 epochs as the loss had reached a sufficiently low value by this point. Experiments were repeated 5 times and a mean percentage of accuracy calculated. There were many random elements at numerous stages of all implementations and so running for multiple iterations and taking the mean of the result accuracy was the only method to ensure that results were legitimately indicative of network performance.

3.4 Splitting the Dataset into Training and Testing

Data was split between train and test sets using leave-p-out cross-validation (LPOCV), meaning that the test set comprised of cases from p users, while all remaining users made up the training set. LPOCV was used due to the real-world constraints of the SelfBACK project, which involves being able to identify users' activities based upon their similarity to other users. This offered an improvement in accuracy over randomly splitting the dataset, though the larger training set caused minor overfitting. Experiments were completed with p set to 5 and 10, to test the effect that increasing test set size had on results.

At run time, the dataset is normalised using standard normal distribution, Equation (4), where μ is the mean of the training set and σ is the standard deviation of the training set. These values were taken from the training set because the test data represented a population of unknown size and distribution.

$$x \in X \mid \frac{x - \mu}{\sigma} \tag{4}$$

3.5 Pair Creation

Pair creation in all experiments was completed after the data had been split into training and test sets to ensure that there was no cross contamination which could effect the results. For pair creation, we defined d as the number of cases in the full dataset and p as the number of cases to be left out for testing. The training set therefore contained $n = d - p$ cases.

Initially, we attempted to exhaustively create all pairs by matching every case with every other case. However, this resulted in the creation of $\sum_1^d$ cases, which made pair formation extremely slow. Instead, two pairs were created for every case in the dataset; a genuine pair (with a random case of the same class) and an impostor pair (with a random case of a different class). This meant that every case was represented at least twice. We enforced equal genuine and impostor pair creation because generating truly random pairs led to an imbalance of more impostor pairs than genuine, at a ratio of approximately 5:1, and had a negative effect on classification. The number of training and test pairs were therefore $2n$ and $2p$ respectively.

4 Results

4.1 Learning Similarity Knowledge with a CSN

The CSN was able to develop good similarity knowledge for all three time and frequency representations of the thigh dataset, though best results were obtained from the thigh DCT data. This is indicated by the percentage of correctly identified pair relationships, which is shown in Table 2.

SelfBACK Thigh Data	Test User Set	Pair Identification Accuracy
Raw	5	93.57
DCT	5	**94.33**
FFT	5	93.87
Raw	10	92.17
DCT	10	**94.30**
FFT	10	93.00

Table 2: CSN Pair Identification Accuracy on the Thigh Dataset

Even the minimum result of 92.17% obtained on the raw thigh dataset using L10OCV demonstrates that more than 92% of test cases have been mapped to appropriate feature vectors. With this in mind, distance between these vectors can act as a proxy for similarity measurements at a case-to-case level. These results support the argument that cases of the same class are grouped closer together within the feature space and lend evidence to the idea that the CSN can be used to form the basis for similarity-based retrieval in a CBR system.

4.2 Comparing a CSN and CNN on the SelfBACK Dataset

As a classifier, the CSN did not perform as well as the CNN, although it acheived over 90% classification accuracy on both experiments. Although the CSN performed competitively on the L5OCV, the CNN displayed much higher accuracy on the L10OCV experiments, as shown in Table 3.

Architecture	Test User Set	Classification Accuracy
CSN	5	90.75
CNN	5	**91.77**
CSN	10	90.03
CNN	10	**92.60**

Table 3: CSN and CNN Comparison on Human Activity Classification

These results indicate that the CSN requires more training data to be able to classify cases than a typical CNN does. The low variance in results across iterations, and resistance to increasing test set size, by both architectures, supports the idea that they generalise well even to large test sets. Although the CSN did not perform as well at classifying cases as the CNN, we argue that the generation of similarity knowledge as a by-product of the classification process is a non-negligible contribution. The main benefit of using the CSN over a traditional CNN implementation is that the output of the CSN produces feature vectors of the original case base which are good representations of how each case fits into the case base as a whole and allows direct, accurate distance measurements as a proxy for measuring similarity between cases.

On reflection, there may be a couple of reasons that the CSN did not perform as strongly on the classification task as the CNN. A more structured method of classifying the output test feature vector, such as exhaustive k-nn sorting or informed class representative selection, could potentially offer better classification results and may be worth further study. It is a distinct possiblity that the class representative which was randomly selected for comparison with the test vector was a poor representative of the class, and that may have influenced the classification of test cases. Exhaustively comparing the test case with all training cases, or using a method of selection to pick class representatives may improve classification accuracy.

5 Conclusion and Further Work

We have demonstrated that a CSN is capable of learning similarity knowledge on the SelfBACK dataset. In addition, we have demonstrated that a CSN can use this similarity knowledge to perform human activity classification on the SelfBACK dataset and can perform competitively with a CNN on this task.

In future work we would like to further explore SNN's capacity to develop simlarity knowledge between cases in order to determine whether this could be used in some manifestation to develop similarity knowledge between features. Our end goal is to use this similarity knowledge in order to populate the similarity arcs which exist between information entities in a Case Retrieval Network (CRN) [6]. If this could be applied, it would offer an inexpensive method to develop efficient coverage of extensive case bases and reduce the initial cost often associated with CRNs in this task. In addition, it would be interesting to explore different methods of pair generation and different methods of utilising the similarity knowledge generated by a CSN for classification.

References

[1] Bell, S. and Bala, K. Lerning Visual Similarity for Product Design with Convolutional Neural Networks. In: Proceedings of the 42nd International Conference and Exhibition on Computer Graphics and Interactive Techniques, SIGGRAPH 2015. Los Angeles, CA, USA. 9 - 13 August 2015. New York, NY, USA: ACM

[2] Berlemont, S., Lefebvre, G., Duffner, S. and Garcia, C. 2015. Siamese Neural Network Based Similarity Metric for Inertial Gesture Classification and Rejection. In: 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2015. Ljubljana, Slovenia. 4 - 8 May 2015. Red Hook, NY, USA: Curran Associates, Inc

[3] Bromley, J., Guyon, I., LeCun, Y., Sackinger, E. and Shah, R. 1993. Signature Verification Using a "Siamese" Time Delay Neural Network. International Journal of Pattern Recognition and Artificial Intelligence, Vol 7(4) pp. 669 - 688.

[4] Chopra, S., Hadsell, R. and LeCun, Y. 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Patter Recognition, CVPR 2005. San Diego, CA, USA. 20 - 25 June 2005. Washington, DC, USA: IEEE Computer Society

[5] Dalal, S., Athavale, V. and Jindal, K. 2011. Case Retrieval Optimisation of Case-Based Reasoning Through Knowledge-Intensive Similarity Measures. International Journal of Computer Applications, Vol 34(3) pp. 12 - 18

[6] Koch, G., Zemel, R. and Salakhutdinov, R. 2015. Siamese Neural Networks for One-Shot Learning. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015. Lille Grand Palais, Lille. 6 âĂŞ 11 July 2015. Red Hook, NY, USA: Curran Associates, Inc

[7] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. 1998. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, Vol 86(11) pp. 2278 - 2324

[8] Lenz, M. and Burkhard, H. D. 1996. Case Retrieval Nets: Basic Ideas and Extensions. In: Proceedings of the 20th Annual German Conference on Artificial Intelligence, KI-96. Dresden, Germany. 17 - 19 September 1996. Berlin, Germany: Springer-Verlag

[9] Sani, S., Wiratunga, N., Massie, S. and Cooper, K. 2016. SELFBACK - Activity Recognition for Self-management of Low Back Pain. In: Proceedings of the 36th SGAI International Conference on Artificial Intelligence, AI-2016. Cambridge, England. 13 - 15 December 2016. Cham, Switzerland: Springer Nature

[10] Stahl, A. 2006. Combining Case-Based and Similarity-Based Product Recommendation. In: Proceedings of the 8th European Conference of Advances in Case-Based Reasoning, ECCBR 2006. Fethiye, Turkey. 4 - 7 September 2006. Berlin, Germany: Springer-Verlag

[11] Stahl, A. and Gabel, T. 2006. Optimising Similarity Assessment in Case-Based Reasoning. In: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI-06. Boston, Massachusetts. 16 - 20 July 2006. Berlin, Germany: Springer-Verlag

[12] Yih, W. et al. 2011. Learning Discriminative Projections for Text Similarity Measures. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, CONLL 11. Portland, OR, USA. 23 - 24 June 2011. Stroudsburg, PA, USA: ACL

Learning Deep Features for kNN-Based Human Activity Recognition

Sadiq Sani, Nirmalie Wiratunga, and Stewart Massie

School of Computing Science and Digital Media,
Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
{s.sani|n.wiratunga|s.massie}@rgu.ac.uk

Abstract. A CBR approach to Human Activity Recognition (HAR) uses the kNN algorithm to classify sensor data into different activity classes. Different feature representation approaches have been proposed for sensor data for the purpose of HAR. These include shallow features, which can either be hand-crafted from the time and frequency domains, or the coefficients of frequency transformations. Alternatively, deep features can be extracted using deep learning approaches. These different representation approaches have been compared in previous works without a consistent best approach being identified. In this paper, we explore the question of which representation approach is best for kNN. Accordingly, we compare 5 different feature representation approaches (ranging from shallow to deep) on accelerometer data collected from two body locations, wrist and thigh. Results show deep features to produce the best results for kNN, compared to both hand-crafted and frequency transform, by a margin of up to 6.5% on the wrist and over 2.2% on the thigh. In addition, kNN produces very good results with as little as a single epoch of training for the deep features.

Keywords: human activity recognition, feature representation, deep learning

1 Introduction

Human activity recognition (HAR) is the computational discovery of human activity from sensor data and is receiving increasing interest in the areas of health care and fitness [3]. This is mainly driven by the need to find innovative ways to encourage physical activity. An example of a health application of HAR is SELFBACK [1] [1], an EU funded project that is developing a self-management system for patients with Lower Back Pain. The motivation for this work is driven by the need for an effective HAR component for SELFBACK, which is required to accurately measure adherence to physical activity targets.

HAR is generally considered as a classification problem where a classifier is trained to identify user activity from sensor data. A CBR approach to this problem makes use of

[1] http://www.selfback.eu/

a kNN classifier in order to facilitate similarity-based reasoning and explanation. However, the effectiveness of a kNN classifier depends on the quality of the feature representation used. Different feature representation approaches have been proposed for HAR, from shallow hand-crafted features to frequency transform features e.g. Fast Fourier Transforms (FFT) and Discrete Cosine Transforms (DCT) coefficients, and more recently, deep learning approaches. All these approaches have had some degree of success and setbacks in performance [6]. It is our view that none of the previous works provides a clear answer to which feature extraction approach is best. Also, previous works have evaluated these feature representation approaches on combinations of different types of data-sets with different mixes of sensor locations and classifiers. In this work, we focus on the feature representation for the kNN classifier using data from two popular body locations, wrist and thigh.

The main contribution of this work is an empirical evaluation of 5 different feature representation approaches across three different classes of features i.e. shallow hand-crafted features, shallow frequency transformation features and deep CNN derived features, for kNN, using sensor data collected from two common body locations, the wrist and the thigh. Wrist data is more prone to random noise compared to data collected at other body locations (e.g. thigh) due to increased variations in movement and posture possible with the hand while undertaking activities. Our goal in this work is to understand which of these feature representations is better suited for the kNN classifier and to analyse any differences in feature performance that may exist between the wrist and thigh.

The rest of this paper is organised as follows: in Section 2, we highlight important related work on feature representation for HAR. Our dataset is described in Section 3. Evaluation is presented in Section 4 and conclusions in Section 5.

2 Related Work on Feature Representation for HAR

Many different feature extraction approaches have been proposed for accelerometer data for the purpose of activity recognition [3]. We broadly classify these into hand-crafted, frequency-transform and deep features.

2.1 Hand-crafted Features

This is the most common approach to HAR and involves the computation of a number of defined measures on either the raw accelerometer data (time-domain) or the frequency transformation of the data (frequency domain) [5]. These measures are designed to capture the characteristics of the signal that are useful for distinguishing different classes of activities. In the case of both time and frequency domains, the input is a vector of real values $\overrightarrow{v} = v_1, v_2, ...v_n$ for each axis x, y and z. A function θ_i is then applied to each vector to compute a single feature value. Typical time domain features include mean, standard deviation and percentiles [10]; while typical frequency domain features include energy, spectral entropy and dominant frequency [2]. The time-domain and frequency domain features used in this work are presented in Table 1

Time Domain Features	*Frequency Domain Features*
Mean	Dominant frequency
Standard deviation	Spectral centroid
Inter-quartile range	Maximum
Lag-one-autocorrelation	Mean
Percentiles (10,25,50,75,90)	Median
Peak-to-peak amplitude	Standard deviation
Power	
Skewness	
Kurtosis	
Log-energy	
Zero crossings	
Root squared mean	

Table 1. Hand-crafted features for both time and frequency domains.

While hand-crafted features have worked well for HAR, a significant disadvantage is that they are domain specific. A different set of features need to be defined for each different type of input data i.e. accelerometer, gyroscope, time-domain and frequency domain. Hence, some understanding of the characteristics of the data is required. Also, it is not always clear which features are likely to work best [5]. Choice of features is usually made through empirical evaluation of different combinations of features or with the aid of feature selection algorithms [9].

2.2 Frequency Transform Features

Frequency transform features extraction involves applying a single function ϕ on the raw accelerometer data to transform this into the frequency domain, where it is expected that distinctions between different activities are more emphasised. The main difference between frequency transform and hand-crafted features is that the coefficients of the transformation are directly used for feature representation without taking further measurements. Common transformations that have been applied include Fast Fourier Transforms (FFTs) and Discrete Cosine Transforms (DCTs).

FFT is an efficient algorithm optimised for computing the discrete Fourier transform of a digital input. Fourier transforms decompose an input signal into its constituent sine waves. In contrast, DCT, a similar algorithm to FFT, decomposes a given signal into it's constituent cosine waves. Also, DCT returns an ordered sequence of coefficients such that the most significant information is concentrated at the lower indices of the sequence. This means that higher DCT coefficients can be discarded without losing information, making DCT better for compression.

For frequency transform feature extraction, a transformation function (DCT or FFT) ϕ is applied to the time-series accelerometer vector $\overrightarrow{v}$ of each axis. The output of ϕ is a vector of coefficients which describe the sinusoidal wave forms that constitute the original signal. Accordingly the transformed vector representations, $\mathbf{x}' = \phi(\mathbf{x})$, $\mathbf{y}' = \phi(\mathbf{y})$ and $\mathbf{z}' = \phi(\mathbf{z})$, are obtained for each axis of a given instance. Additionally we derive a further magnitude vector, $\mathbf{m} = \{m_{i1}, ..., m_{il}\}$ of the accelerometer data for each instance as a separate axis, where m_{ij} is defined as $m_{ij} = \sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2}$. As with $\mathbf{x}'$, $\mathbf{y}'$ and $\mathbf{z}'$, we also apply ϕ to $\mathbf{m}$ to obtain $\mathbf{m}' = \phi(\mathbf{m})$. The final feature representation is obtained by concatenating the absolute values of the first l coefficients of $\mathbf{x}'$, $\mathbf{y}'$, $\mathbf{z}'$ and $\mathbf{m}'$ to produce a single feature vector of length $4 \times l$. The value $l = 80$ is used in this work, which is determined empirically. Further information on feature representation using DCT and FFT can be found in [7].

2.3 CNN Feature Extraction

Convolutional Neural Networks (CNNs) have been applied for feature extraction in HAR, due to their ability to model local dependencies that may exist between adjacent data points in the accelerometer data [8]. CNNs are a type of Deep Neural Network that is able to extract increasingly more abstract feature representations by passing the input data through a stack of multiple convolutional operators [4], where each layer in the stack takes as input, the output of the previous layer of convolutional operators. An example of a CNN is shown in Figure 1.

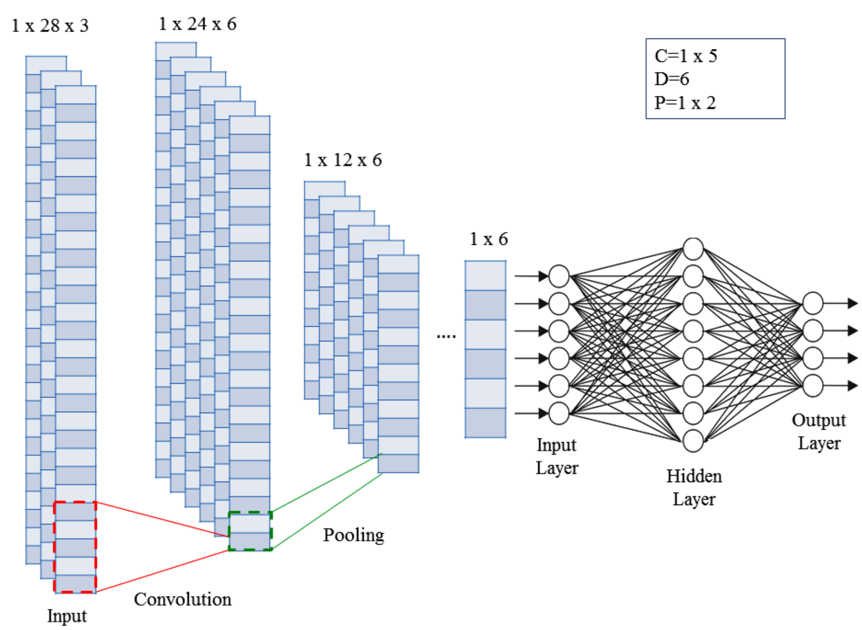


Fig. 1. Illustration of CNN

The input into the CNN in Figure 1 is a 3-dimensional matrix representation with dimensions $1 \times 28 \times 3$ representing the width, length and depth respectively. Tri-axial accelerometer data typically have a width of 1, a length l and a depth of 3 representing the

x, y and z axes. A convolution operation is then applied by passing a convolution filter over the input which exploits local relationships between adjacent data points. This operation is defined by two parameters, D representing the number of convolution filters to apply and C, the dimensions of each filter. For this example, $D = 6$ and $C = 1 \times 5$. The output of the convolution operation is a matrix with dimensions $1 \times 24 \times 6$, these dimensions being determined by the dimension of the input and the parameters of the convolution operation applied. This output is then passed through a Pooling operation which basically performs dimensionality reduction. The parameter P determines the dimensions of the pooling operator which in this example is 1×2, which results in a reduction of the width of its input by half. The output of the pooling layer can be passed through additional Convolution and Pooling layers. The output of the final Pooling layer is then flattened into a 1-dimensional representation and then fed into a fully connected neural network. The entire network (including convolution layers) is trained through back propagation over a number of generations until some convergence criteria is reached. Detailed description of CNNs can be obtained in [4].

Note that once the CNN is fully trained, it can used to provide feature representations for use with other types of classifiers e.g. kNN. This is achieved by cutting off the trained network after the final pooling layer and just before the fully-connected neural network. Each training example is then passed through the convolutional network in order to obtain an abstract representation which is used to train the kNN classifier. A similar operation is performed for each test example to obtain an abstract representation which is passed to kNN for classification.

3 Dataset

A group of 34 volunteer participants was used for data collection. The age range of participants is 18 - 54 years and the gender distribution is 52% Female and 48% Male. Data collection concentrated on the activities provided in Table 2.

Activity	**Description**
Walking	Walking at normal pace
Jogging	Jogging on a treadmill at moderate speed
Up Stairs	Walking up 4 - 6 flights of stairs
Down Stairs	Walking down 4 - 6 a flights of stairs
Standing	Standing relatively still
Sitting	Sitting still with hands on desk or thighs

Table 2. Details of activities classes in our dataset.

This set of activities was chosen because it represents the range of normal daily activities typically performed by most people. Data was collected using the Axivity Ax3 tri-axial accelerometer [2] at a sampling rate of 100Hz. Accelerometers were mounted

[2] http://axivity.com/product/ax3

on the right-hand wrists and right thighs of the participants. Activities are evenly distributed between classes as participants were asked to do each activity for the same period of time (3 minutes).

4 Evaluation

Evaluations are conducted using a leave-one-person-out methodology where each user's data is held out for testing in turns, while the remaining 33 are used for training. In this way, we are testing the general applicability of the system to users whose data is not included in the trained model. Performance is reported using macro-averaged F1 and kNN is used for classification with euclidean distance and the parameter $k = 5$.

The representations included in our comparison are as below:

- *Time*: Time domain hand-crafted features
- *Freq*: Frequency domain hand-crafted features
- *DCT*: DCT frequency features
- *FFT*: FFT frequency features
- *CNN*: CNN deep features with soft-max classifier
- *CNN-kNN*: CNN deep features with kNN classifier

For the CNN, after experimenting with different parameter settings, the final configuration used for thigh data had 3 convolution layers with 150, 100 and 80 convolution filters respectively. The configuration used for wrist data had 5 convolution layers with the same numbers of convolution filters as the thigh data in the first 3 layers and 60 and 40 convolution filters in the fourth and fifth layers respectively. Each convolution layer was followed by a max pooling layer. A convolution filter of size 10 and pooling size of 2 were used on all convolution and pooling layers respectively. The last pooling layer is connected to a fully connected network with 2 hidden layers, where the first layer had 900 units and second layer had 200 units. A dropout probability of 0.5 was used for each hidden layer. The final output layer had 6 units representing the 6 activity classes in our dataset and uses soft-max regression. Loss is computed using cross-entropy and the network is trained using back-propagation for 200 epochs.

4.1 Results

Results of our comparative evaluation are shown in Figure 2. The best results for both thigh and wrist are achieved using deep features (CNN and CNN-kNN). This highlights the fact that kNN, using deep features, can rival the performance of state-of-the-art deep learners, while still providing the ability for similarity-based reasoning and explainability that makes kNN desirable. In general, HAR performance is higher using thigh data compared to wrist by a margin of up to 14.7% (for DCT). This indicates that the thigh is a much better position for HAR compared to the wrist. However, the benefit from deep feature representations is consistent on both wrist and thigh.

Out of the shallow features (Time, Freq, DCT and FFT), the best performance is achieved using DCT. This is consistent with our previous findings [7]. However, in comparison with DCT, CNN-kNN produces 6.5% and 2.2% improvement on the wrist and thigh respectively. Both improvements are statistically significant at 95% using a paired t-test.

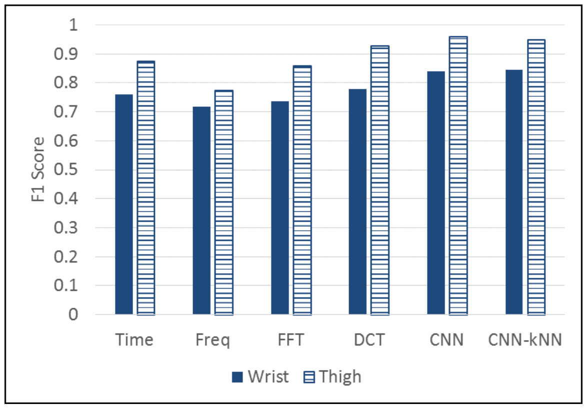


Fig. 2. Results of different representations(Time, Freq, DCT, FFT, CNN, CNN-kNN)

It is known that one of the major bottleneck of applying deep learning is the amount of time required for training. Hence, it is important to understand the effect of training time on the performance of both CNN and CNN-kNN. Particularly, we would like to see the level of performance that can be achieved with minimum training time. Figure 3 presents the results of CNN and CNN-kNN at between 1 to 5 epochs of training for the wrist (left) and thigh (right).

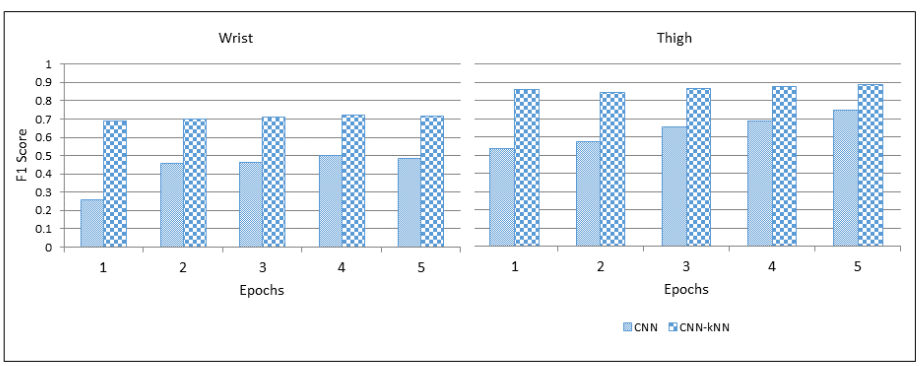


Fig. 3. Results for CNN and CNN-kNN after training for between 1-5 epochs.

Note that CNN-kNN outperforms CNN on both wrist and thigh at all 5 training epochs. Also, the performance of CNN-kNN is good (on par with Freq) even after a single epoch of training. These results are an important finding of this work and demonstrate the robustness of kNN in effectively using deep features, irrespective of the amount of time spent on training.

Finally, we analyse the effect of the depth of our network on the quality of deep features we are able to extract for kNN. Figure 4 shows the performance of CNN-kNN with different numbers of convolution layers between 3 and 5. Note that the best performance for the thigh is achieved using 3 convolution layers (0.949) and performance gradually decreases with the addition of more convolution layer (0.947 for 4 and 0.937 for 5). In contrast, performance on the wrist produces a significant increase (at 95% using a paired t-test) with additional layers from 0.73 for 3 layers to 0.84 for 5 layers. This indicates that deeper layers are required for effective feature extraction on more difficult datasets. However, a relatively shallow architecture seems sufficient for easier datasets.

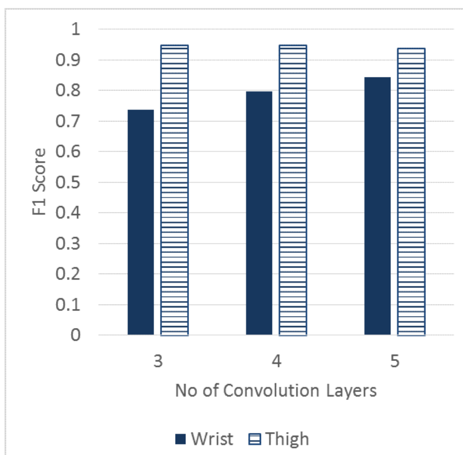


Fig. 4. Results for CNN-kNN at different depths between 3-5 convolution layers.

5 Conclusion

In this paper, we have presented an analysis of different feature representation approaches for the purpose of human activity recognition using kNN. These feature representation approaches can be broadly categorised into three classes: handcrafted, frequency transform and deep features. Evaluation is conducted using accelerometer data collection from two different body locations: wrist and thigh. Results show deep features to significantly out-perform the other representation types on both wrist and thigh by a margin of over 6.5% on the wrist, and 2.2% on the thigh. In addition, our evaluation shows kNN to be very effective at using deep features, even when a minimum amount of time spent in training these deep features.

Future work will investigate the use of RNN for feature extraction due to their ability to model the sequential relationship inherent in the time series accelerometer data.

References

1. K. Bach, T. Szczepanski, A. Aamodt, O. E. Gundersen, and P. J. Mork. Case representation and similarity assessment in the selfback decision support system. In *Proceedings of 24th International Conference on Case-Based Reasoning, ICCBR 2016*, pages 32–46. Springer International Publishing, 2016.
2. D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.
3. O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *Communications Surveys & Tutorials, IEEE*, 15(3):1192–1209, 2013.
4. Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 255–258. MIT Press, 1998.
5. T. Plötz, N. Y. Hammerla, and P. Olivier. Feature learning for activity recognition in ubiquitous computing. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, IJCAI'11, pages 1729–1734. AAAI Press, 2011.
6. D. Ravi, C. Wong, B. Lo, and G. Z. Yang. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE Journal of Biomedical and Health Informatics*, 21(1):56–64, Jan 2017.
7. S. Sani, N. Wiratunga, S. Massie, and K. Cooper. Selfback-activity recognition for self-management of low back pain. In *Research and Development in Intelligent Systems XXXIII: Incorporating Applications and Innovations in Intelligent Systems XXIV*, pages 281–294. Springer, 2016.
8. M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *Proceedings of 6th International Conference on Mobile Computing, Applications and Services*, pages 197–205, 2014.
9. S. Zhang, P. Mccullagh, and V. Callaghan. An efficient feature selection method for activity classification. In *Proceedings of IEEE International Conference on Intelligent Environments*, pages 16–22, 2014.
10. Y. Zheng, W.-K. Wong, X. Guan, and S. Trost. Physical activity recognition from accelerometer data using a multi-scale ensemble method. In *IAAI*, 2013.

Data driven case base construction for prediction of success of marine operations

Bjørn Magnus Mathisen, Agnar Aamodt, Helge Langseth

Norwegian University of Science and Technology
bjornmm@ntnu.no, agnar@ntnu.no, helge.langseth@ntnu.no

Abstract. It is a common situation to have lots of recorded data that you want to use for improving a process in your organization or make use of this data to provide new services or products. Starting with one primary data set we describe a system that enhances this data set to a level such that it can be used by a deep learning system. This deep learning system then creates a model based on this data set, trying to predict operational windows for marine operations. Using this model the system extracts cases for use in a CBR-system aimed at providing operational support. This paper describes the partial implementation and results of this system.

Keywords: Data Science, Deep Neural Networks, Data Analytics, Case-based Reasoning

1 Introduction

Critical operations are often meticulously planned and subject to many parameters that decide if and how these operations are performed. Some of these parameters are called operational time windows, which in marine environments often are connected to external factors such as weather.

This paper uses machine learning to predict favorable operational time windows or warn of unfavorable operational windows, so that critical operations can be planned with better accuracy, e.g. when the operation should ideally take place. One way of doing this is to look at historical data of previously executed operations. By combining data on successful and unsuccessful operations with the relevant context of that operation, we create a data set that can be used to find indicators for success or failure in advance. Which context that is relevant is dependent on the nature of operational window; wind and fog are important contexts for aviation, while waves and current are important for marine operations but not aviation.

This paper focuses on marine operations, and we analyze event data captured from boats moving in and out of zones connected to aquaculture installations. Next, we calculate the duration of these events and connect them to the relevant context and the associated success or failure classification.

The data used in this analysis is gathered as part of the EXPOSED project[1]. This project aims to develop enabling- and applied technologies for exposed

[1] http://exposedaquaculture.no/en/

aquaculture operations. The work we describe aims to improve planning of operations on aquaculture installations on exposed locations.

The data is a subset of boats moving across geofences attached to aquaculture installations. This system consists of two zones around every aquaculture installation in Norway: One outer zone 400 meters from the outer points of the structures holding the fish themselves (not including the control building/fishfeed silos). The inner zone is 20 meters from the structure. These limits are in adherence to government regulations that no boat should fish within the outer zone and no boat should move within the inner zone unless the boat is there to operate on the installation.

An example of geofencing zones are shown in Fig. 1 below.

Fig. 1: The Green line show the outer geofence zone, the red line shows the inner geofence zone.

An event is created each time a boat crosses any of the geofence zones, marking the time. Table 1 below shows an example of a typical event.

Event ID	Location-ID	Vessel Name	Time	LocationZone	EventType
81766	12966	Vessel A	2014-09-02 21:39:32	1	1
81767	12966	Vessel A	2014-09-02 21:40:11	1	2

Table 1: This table shows an example of two events of a vessel entering (EventType=1) and leaving (EventType=2) the outer zone (LocationZone=1) of location 12966.

In data gathered in the EXPOSED project, the aquaculture industry reports on several possible problems with fish feed carriers interacting with aquaculture installations: Approaching the feed barges, often placed in shallow waters; Knowing which barge container to fill with what feed; Planning according to weather and route to enable the installation crew to attend the operation; And the fact that impact and currents from the boat can damage the installation.

As our data only gives us the time spent in two different proximities to the aquaculture installation there will be limits to which types of operational problems we can detect, and it will be very hard to discern between different causes (other than bad weather which is very general) of any detected problem.

The architecture of the full decision support system for EXPOSED is illustrated in Fig. 2. In this paper we only present results from parts of the system. Future work will integrate these results with the other modules (e.g. knowledge models) to complete the system to a state where it can be verified in the field.

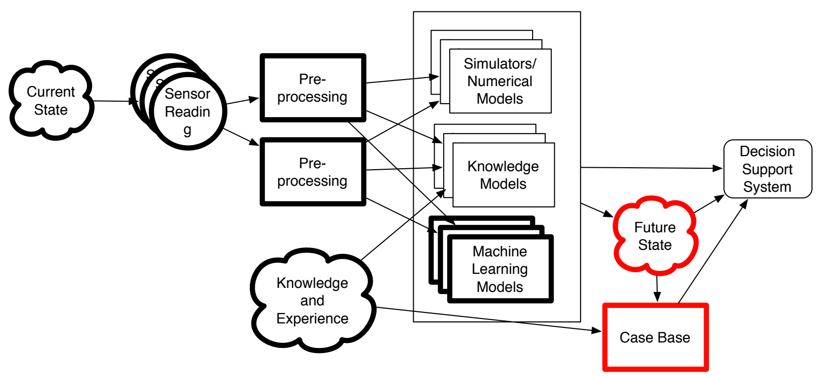


Fig. 2: The architecture of the planned systems. The parts implemented are highlighted, the case base and the future state is highlighted in red as being the current target for development.

Our main hypothesis is that given enough contextual weather data a deep neural network should be able to predict the length of a maritime operation at a aquaculture installation, enabling us to predict favorable operational windows. The main contribution of this paper is to show the reader the process of gathering, collating, filtering of data and subjecting this data to an analysis.

This paper is structured as follows; Section 2 introduces related work and our work in the light of this previous work. Section 3 describes the methods used in our work as well as the data sources used. Section 4 shows the result of our experiments, while section 5 presents the conclusion along with a discussion of the results.

2 Related work

In this work we aim to extract cases from a time series of events, CBR research has been done on several aspects of automatic case-authoring.

In CBR there has been a lot of focus on how to measure competence and utility of a case-base [1,2]. In [3], they do this via reversing deletion policies constructed in [4] that try to improve case base utility without degrading competence.

Several works [5,6,7] use NLP to extract cases from structured and unstructured ([8,9,10]) text.

More specifically connected to the task of extracting cases from time series is the work done by Bach et.al. [11] where they employ clustering of time-series events in time and space, in combination with other detection methods. Funk et. al [12] uses different models of how predictive (or discriminatory) different time-series patterns are to different medical diagnosis of stress. For more insight into work done in time-series analysis connected to CBR research we suggest chapter 3.3 in [13]

The work presented in this paper shares the approach of Bach et al. [11] in that we try to extract the useful data points from the time series via clustering and filtering. Our work differs from the previous work in that we have very few verified cases apriori or during learning. In other words, the time-series is in all practical sense unlabeled for our use. We will try to apply common knowledge about how long an operation usually takes to perform. Then we can extract failed operations from the even time series to create cases that exemplify failed operations.

3 Method

To enable the deep learning system to correctly model and predict the time spent at an installation, we need to provide it with as much context data as possible for each of the event data points. In addition, we need the data to be as noise free as possible, thus we want to filter away operations that naturally have a high degree of variation in time spent at the location. We address these two requirements by combining the primary data set with other data sets, to enable us to provide filtering and context. An illustration of this process can be seen in Fig. 3 . Below we describe each of the data sets.

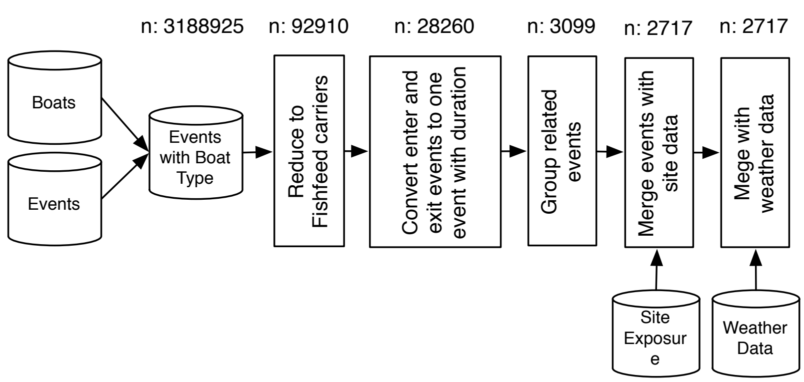


Fig. 3: This figure illustrates how the different data sources are combined and filtered to provide the deep learning system as much context as possible.

Boat data set As mentioned in the introduction we do not want to analyze all the traffic data of all of the boats. To verify that our method is usable in at least one instance, we want to look at a specific type of boat that has stable characteristics when it comes to the parameters (e.g. time and stability of time) of the operations it executes on the installation. We chose fishfeed boats in this case, as they only do one type of operation. That way we do not need to deduce the type of operation from the event data (one less hidden variable). In addition, this operation should be stable in the time it takes to execute it. To filter the data accordingly we need to combine the event data set with a data source that describes the boats. We can then easily extract the fishfeed boats.

NORA10 data set NORA10 [14,15] is a data set that describes output of a precise weather model (hind-cast), that is validated by measurements. It has a higher resolution (10km) than most other models (e.g. the much used ERA[2] model with 80km resolution) as it is re-sampled for this specific region around Norway. We sample this model for each of the installations and at each time of each event (in the case of long events we use the median time of the event). We sample every datatype that we think will have an impact on the time spent on an operation: wind speed, wave direction, wind direction, significant swell wave height and significant wave height.

Exposure data set SINTEF EXPOSED has produced a data set [16] that describes the degree of exposure for a large number of the installations that are used in the event data set. This data set provides a level of exposure for 360 degrees around the installation (from 0 to max, where max is no land in sight). We combine our weather data with this (described above), thus we combine the wind direction of the wind with how exposed the location is in the direction of the wind using a filter that combines exposure level from +/- 10 degrees around the direction of the wind.

3.1 Extracting time spent in zones.

The data set needs to contain the time spent in the zones around the aquaculture installations. The raw data only contains events of entering and exiting the zones. To extract this we sequentially find each exit from a zone then search backwards for the entry to that zone by the same boat, then compute the time spent in that zone.

3.2 Grouping events close in time

After converting all discrete events into events with a duration, we still ended up with a lot of extremely short events. This is most probably caused by boats trying to stay close to the installation but the dynamic positioning system moves them in and out of the inner or outer zones. To counter this fact we grouped all events with the same boat at the same location within 1 hour into one event. However, after this grouping there is still 63% (or 244) of the events within the first 10 minute window. These are events within a zone that is less than ten minutes in duration and without another event in the same location within one hour of the original event. There are three possible explanations for these strange events: 1. The boat is passing through the location, and not returning for at least one hour. Or otherwise briefly enters and exists the zone, without this fact having any effect on the operation. 2. The boat tries to perform an operation at the location but has to abort and leaves within ten minutes. 3. The event was not registered correctly when the data was gathered. The most probable cause for most of these events are boats that travel through the zone heading for another location. This hypothesis can be tested by removing outer zone events from the distribution. As the inner zone is small, very few of these big fishfeed carrier boats would drive through the inner zone of an aquaculture installation when

[2] `http://www.ecmwf.int/en/research/climate-reanalysis/era-interim`

heading somewhere else. We can still see 244 events that are of duration 10 minutes or less within the inner zone of an aquaculture installation. Figure 4 looks at the 1 minute distribution within the first 10 minutes to try to find the causes for the high number of short stay events. And once again we can see that many of the events are very short, with very few events lasting more than 3 minutes. This further supports our first hypothesis.

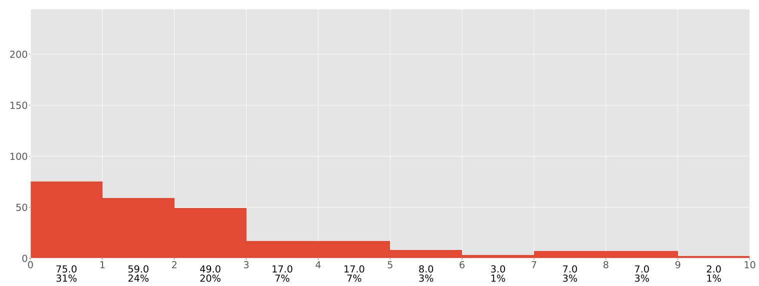


Fig. 4: Distribution of events over length of stays in all inner zone after grouping all events within a 1 hour time window. Zoomed into the first 10 minutes.

One problem with our approach so far is that some events are very far apart in time as well as having different zone types. One example being one boat having a 0 second stay in the inner zone of location 31437 at 18:23 the 28th of November, however the boat entered the outer zone of the same site at 17:04 the same day, and exited zone 1 of that location at 18:24. We can then conclude that the boat spent approximately 1 hour and 20 minutes at the location in the outer zone, then very briefly entered the inner zone before leaving the location. Again supporting the first hypothesis. From this we can see that including inner zone in analyzing fishfeed carrier operations adds very little information to our analysis as the fishfeed carriers do not enter the inner zone when transferring fishfeed. As a consequence we discard the inner zone data. We are still left with 2401 events with a duration shorter than 10 minutes. Fig 5 shows the distribution of these events length in stay. We can see that most of these are shorter than 5 minutes, and most probably does not represent actual maritime operations (or failed tries), but rather traveling through the zone. Thus we discard events shorter than 10 minutes, giving us the final distribution shown in Fig. 6.

3.3 Predicting the operational time using Deep Learning

To extract cases that exemplify instances where the weather conditions stops a fishfeed operation from being successfull, we are currently building a deep learning model aimed at predicting the time spent at the installation, with the given weather and level of exposure at the time and location. The input to the model is: draft and length of the boat, wind speed[3], distance between the model grid point and actual site coordinate, wave direction[3], wind direction[3], maximum level of exposure at location, significant swell wave height[3], month, hour, wind effect (wind speed combined with

[3] Measured at the closest grid point in NORA10

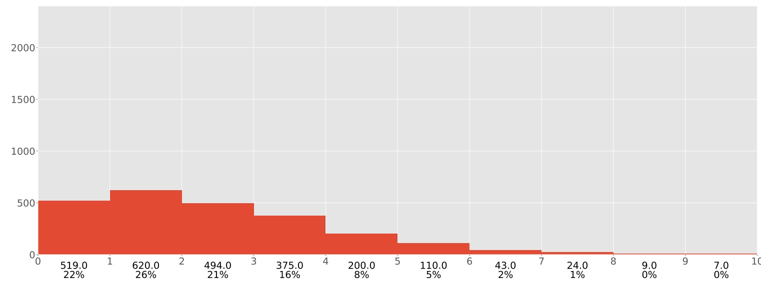


Fig. 5: Distribution of events over length of stays in all outer zone after grouping all events within a 1 hour time window. Zoomed into the first 10 minutes.

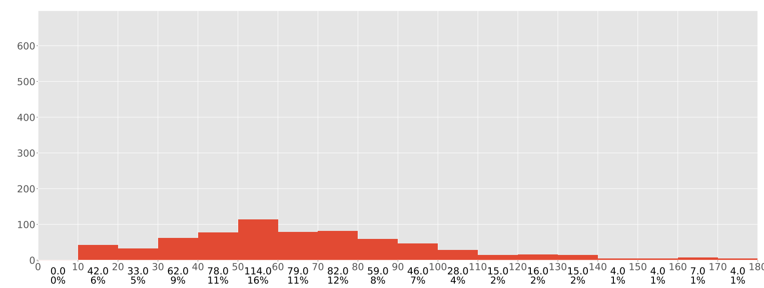


Fig. 6: Distribution of events over length of stays in all outer zone after grouping all events within a 1 hour time window. With all stays smaller than 10 minutes removed.

exposure levels in the wind direction +/- 10 degrees) and significant wave height. The output of the model is the amount of time spent on the installation.

The regression was implemented using python. We used sklearn for preprocessing and scaling (MinMax scaling) of input data (including regression target). The Keras library for deep learning was used for the regression itself, with a input layer of $inputcolumns + 1 = 14$ nodes. We used 3 hidden layers with 13 nodes each and a output layer of 1 node. All nodes used the ReLU activation function.

4 Results

The current results show that there is little information in the gathered data (through the NORA10 model and exposure levels) that account for the variance shown in the time spent at the locations. The neural network models presented in the previous Section 3.3 gets very low accuracy (0.11%, which means the predictor is very slightly better than just outputting the average) in terms of predicting how long a fish feed boat stays at a aquaculture installation. Figure 7 shows the length of all of the events in the chronologically in blue and the predicted length in orange. The "Time Spent" axis is normalized values of the time spent in near a installation where $y = 1.0$ represents the longest stay recorded in the training data. There are obvious differences between predicted and true values; predicted values consistently returns too high values, and fails to predict short stays. A cross validated ($cv = 5$) hyper parameter grid search was performed and showed no better performance at 10 hidden layers with 56 nodes in each hidden layer.

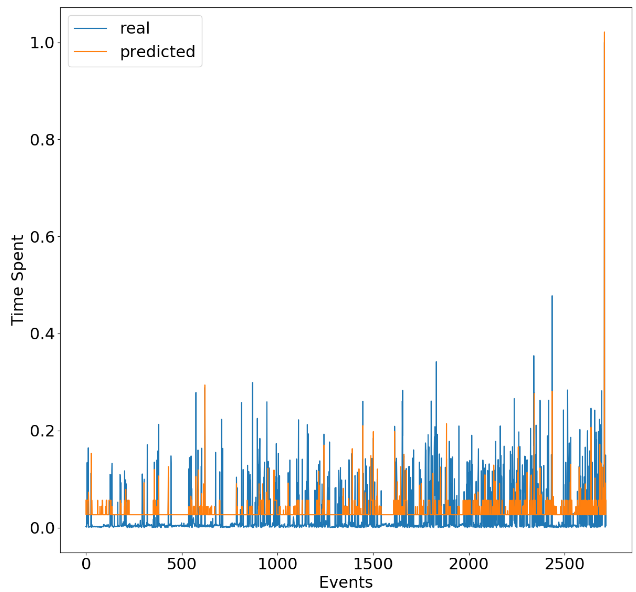
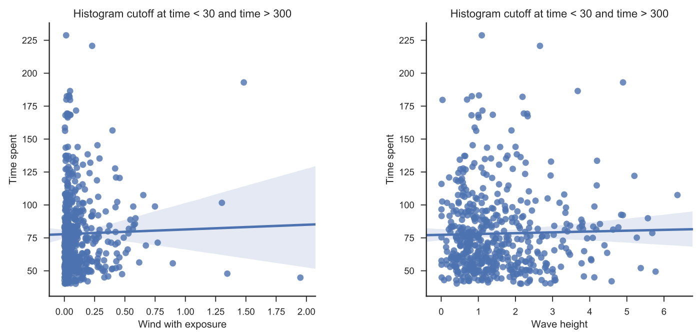


Fig. 7: This shows the DNN model try to predict the amount of time spent at a installation in orange, and the actual time spent in blue. The X-axis is simply the record number, where the record are ordered along the time axis.

After we received the disappointing results we created scatter plots of two weather variables in relation to the length of stay at the installations. Typically most would assume there would be a pattern of some correlation between the weather and the length of stay. However Figure 8 shows that neither wind (8a) or waves (8b) reveals any obvious correlation patterns against time spent at installations.

In addition we did a principal component analysis of the data, to discover if there where any clear principal components that could contain the variance in the data. The components returned: $C = (0.127, 0.117, 0.109, 0.099, 0.091, 0.039, 0.034, 0.028, 0.020, 0.011, 0.008, 0.004, 0.002, 0.000)$ Where the sum of components $sum(C) =$ 0.6967 indicating that the total of the components could account for little of the variance. Finally we tried a standard method for non-linear regression as a base-line result to measure the DNN against. We tried Epsilon-Support Vector Regression (SVR) which scored with a coefficient of determination $R^2 = -0.83$ which is worse than constantly predicting the



(a) Wind correlation (b) Wave correlation

Fig. 8: Scatter plot illustrating correlation between the weather and the time spent at the installation

mean of the target (which would give $R^2 = 0.0$). This final result shows in the context of the other results listed above us that the data set may not contain the features needed to predict the length of the stay at a installation.

5 Conclusions and future work

We started the work with a hypothesis that whether or not a fishfeed boat operation (loading of fishfeed from boat to barge) succeeded depended on the weather, and that such a failure could be detected from the length of time the fishfeed boat stayed at the aquaculture installation. Our analysis did not find any deterministic correlation between the weather and location data and the length of the stay at the installation. There can be many reasons for this, we will try to list some of the reasons we think are probable;

The first possibility is that despite our efforts to remove noise from the data, the data still contains noise. This includes the three factors listed in the introduction section and other possibilities we have not considered.

Second, given the size of the boats and their stability, they can operate during harsh conditions. In addition these boats are expensive in operation, and even more expensive if they fail to deliver feed at the appointed time, possibly starving the fish at the installation. Thus these boats are already subject to careful operational planning. It may therefore be that there is none to very few failed fishfeed operations in the data captured. An additional consequence is that the time spent during operations has very low variance.

Extending this work would start with confirming these possible explanations for the lack of correlation found in our data. We would also like to gather further data, extending the number of events beyond the current 2700. This would enable us to train and test our models with more rigor and less uncertainty.

6 Acknowledements

None of the work done in this paper would have been possible without the support of the EXPOSED project. Special thanks to ANTEO (`http://anteo.no/`) for providing data to this experiment and working with us to make use of this data.

References

1. Barry Smyth and Elizabeth McKenna. *Modelling the competence of case-bases*, pages 208–220. Lecture Notes in Computer Science. Springer Nature, 1998.
2. Barry Smyth and Elizabeth McKenna. *Building Compact Competent Case-Bases*, pages 329–342. Case-Based Reasoning Research and Development. Springer Nature, 1999.
3. Jun Zhu and Qiang Yang. Remembering to add: competence-preserving case-addition policies for case-base maintenance. In *IJCAI*, volume 99, pages 234–241, 1999.
4. B Smyth and M Keane. Remembering to forget: A competence-preserving deletion policy for cbr. In *Proceedings IJCAI-95*, 1995.
5. Chunsheng Yang, Benoit Farley, and Bob Orchard. Automated case creation and management for diagnostic cbr systems. *Applied Intelligence*, 28(1):17–28, Feb 2007.
6. Qiang Yang and Hong Cheng. Case mining from large databases. *Lecture Notes in Computer Science*, page 691–702.
7. Marvin Zaluski, Nathalie Japkowicz, and Stan Matwin. Case authoring from text and historical experiences. *Lecture Notes in Computer Science*, page 222–236, 2003.
8. Kerstin Bach, Klaus-Dieter Althoff, Régis Newo, and Armin Stahl. *A Case-Based Reasoning Approach for Providing Machine Diagnosis from Service Reports*, pages 363–377. Case-Based Reasoning Research and Development. Springer Nature, 2011.
9. Valmi Dufour-Lussier, Florence Le Ber, Jean Lieber, and Emmanuel Nauer. Automatic case acquisition from texts for process-oriented case-based reasoning. *Information Systems*, 40(nil):153–167, 2014.
10. Benoit Farley. From free-text repair action messages to automated case generation. In *Proceedings of AAAI 1999 Spring Symposium: AI in Equipment Maintenance Service & Support, Technical Reprot SS-99-02, Menlo Park, CA, AAAI Press*, pages 109–118, 1999.
11. Kerstin Bach, Odd Erik Gundersen, Christian Knappskog, and Pinar Öztürk. Automatic case capturing for problematic drilling situations. In *International Conference on Case-Based Reasoning*, pages 48–62. Springer, 2014.
12. Peter Funk and Ning Xiong. Case-based reasoning and knowledge discovery in medical applications with time series. *Computational Intelligence*, 22(3-4):238–253, Aug 2006.
13. Odd Erik Gundersen. *Enhancing the Situation Awareness of Decision Makers by Applying Case-Based Reasoning on Streaming Data*. PhD thesis, NTNU, 2014.
14. Øyvind Breivik, Magnar Reistad, and Hilde Haakenstad. A high-resolution hindcast study for the north sea, the norwegian sea and the barents sea. In *10th International Workshop on Wave Hindcasting and Forecasting*, 2007.
15. Magnar Reistad, Øyvind Breivik, Hilde Haakenstad, Ole Johan Aarnes, Birgitte R Furevik, and Jean-Raymond Bidlot. A high-resolution hindcast of wind and waves for the north sea, the norwegian sea, and the barents sea. *Journal of Geophysical Research: Oceans*, 116(C5), 2011.
16. Pål Lader, David Kristiansen, Morten Alver, Hans. V Bjelland, and Dag Myrhaug. Classification of aquaculture locations in norway with respect to wind wave exposure. In *Proceedings of the ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering OMAE2017*, 2017.

3rd Workshop on PROCESS-ORIENTED CASE-BASED REASONING PO-CBR@ICCBR-2017

June 26, 2017, Trondheim, Norway

Miltos Petridis[1], Mirjam Minor[2], Stefania Montani[3], Odd Erik Gundersen[4]

[1] Middlesex University London, UK
[2] Goethe University, Frankfurt, Germany
[3]University of Piemonte Orientale, Italy
[4]NTNU, Trondheim, Norway

[1]m.petridis@mdx.ac.uk,[2]minor@informatik.uni-frankfurt.de,
[3]stefania.montani@unipmn.it, [4]odderik@idi.ntnu.no

Keywords: Process Oriented, Case-Based Reasoning.

1 Workshop overview

The third workshop on Process-oriented Case-based Reasoning (PO-CBR) is dedicated to address the challenges of integrating Case-Based Reasoning (CBR) with process-oriented research areas like Business Process Management, Workflow Management and Planning. Consequently, several types of processes are in the interest of this workshop, mainly business processes, software processes, planning processes, and search processes.

Business Process Management (BPM) is a set of activities aimed at defining, executing, monitoring and optimizing BP, with the objective of making the business of an enterprise as effective and efficient as possible, and of increasing its economic success. Such activities are highly automated, typically by means of the workflow technology. BPM activities, and BP optimization in particular, may ask the enterprise to be able to flexibly change and adapt the predefined process schema, in response to expected situations (e.g. new laws, reengineering efforts) as well as to unanticipated exceptions and problems in the operating environment (e.g. emergencies).

The agile workflow technology is the technical solution which has been invoked to deal with such adaptation and overriding needs. In order to provide an effective and quick workflow change support, many agile workflow systems share the idea of recalling and reusing concrete examples of changes adopted in the past. To this end, Case-based Reasoning (CBR) has been proposed as a natural methodological solution.

Software Processes can be studied from different points of view. In the Software Engineering field, artifacts like models, diagrams, etc. define Software Development Processes that can be reused to generate new applications. There is also a significant trend on reusing different software components to compose a workflow that models the behavior of a system. Web Services, Scientific Software or Product Lines are some examples of such approaches. In this topic we cannot forget a closely related domain like Planning. All these areas are related to Software Processes and can take advantage of the CBR paradigm to reuse existing solutions, components, compositions or plans.

As a matter of fact, in recent years many examples of CBR-based process change reuse and workflow adaptation support have been proposed in the literature. The workshop should serve as a means for exchanging novel as well as more consolidated ideas and examples in the field, and to identify promising research lines and challenges for the future. Furthermore, the automatic monitoring and anomaly detection (real time or retrospective) in Business processes, including the automation of decision support on Business Process execution and design is becoming important. Finally, the reuse of knowledge across Business processes and workflows, cold start challenges and case base maintenance are other challenges in this area.

This year the PO-CBR workshop also hosts two papers focusing on health-care applications.

1.1 Key workshop areas of interest

The Key areas of interest for the workshop are:

- Methodological issues:
  - Case-based representation of process knowledge (by workflows, traces, plans, etc.)
  - Case-based retrieval for process optimization
  - Similarity measures for process optimization
  - Experience reuse in PO-CBR
  - Case-based adaptation for process optimization
  - Extraction of process knowledge
  - Visualization and explanation of process knowledge
  - Cross-process knowledge reuse
  - Maintenance of business process knowledge
  - process-oriented transfer learning

- Applications, systems and tools:
  - PO-CBR applications in Business Process Management, Software Processes, E-Science, Web Science, E-Governance, E-Health, product development, search, games, cooking, and further application domains
  - Evaluating CBR tools for PO-CBR
  - Agile workflow technology with CBR components
  - CBR in (commercial) workflow management tools
  - Applications of PO CBR in Health
- Lessons learned in PO-CBR investigations
- Challenge tasks for CBR systems in the context of business processes, software processes, planning processes, search processes, monitoring processes and decision support processes

1.2 Workshop Organisation

Workshop Organisers

- Miltos Petridis, University of Middlesex, London, UK
- Mirjam Minor, Goethe University, Frankfurt, Germany
- Stefania Montani, University of Piemonte Orientale, Italy
- Odd Erik Gundersen, NTNU, Norway

Programme committee

- Klaus Dieter Althoff, DFKI / University of Hildesheim, Germany
- David Aha, Naval Research Lab, USA
- Ralph Bergmann, Uninersity of Trier, Germany
- Isabelle Bichindaritz, State University of New York at Oswego, USA
- Juan Manuel Corchado, University of Salamanca, Spain
- Pedro A. González-Calero, Complutense University of Madrid, Spain
- Stelios Kapetanakis, University of Brighton, UK
- David Leake, Indiana University, USA
- Beatriz Lopez, University of Girona, Spain
- Jixin Ma, University of Greenwich, UK
- Cindy Marling, Ohio University, Athens, USA
- Hector Muñoz-Ávila, Lehigh University, USA
- Luigi Portinale, Universita' del Piemonte Orientale "A. Avogadro", Italy
- Juan A. Recio-Garcia, Universidad Complutense de Madrid, Spain
- Rainer Schmidt, University of Rostock, Germany
- Barbara Weber, University of Innsbruck, Austria

Automatic Selection of Optimization Algorithms for Energy Resource Scheduling using a Case-Based Reasoning System

Ricardo Faia[1], Tiago Pinto[1,2], Tiago Sousa[1], Zita Vale[1] and Juan Manuel Corchado[21]

[1]GECAD – Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Institute of Engineering, Polytechnic of Porto (ISEP/IPP), Porto, Portugal
{rfmfa, tabsa, tmcfp, zav}@isep.ipp.pt

[2]BISITE Research Centre, University of Salamanca (USAL), Calle Espejo, 12, 37007 Salamanca, Spain
{tpinto, corchado}@usal.es

Abstract. This paper proposes a case-based reasoning methodology to automatically choose the most appropriate optimization algorithms and respective parameterizations to solve the problem of optimal resource scheduling in smart energy grids. The optimal resource scheduling is, however, a heavy computation problem, which deals with a large number of variables. Moreover, depending on the time horizon of this optimization, fast response times are usually required, which makes it impossible to apply traditional exact optimization methods. For this reason, the application of metaheuristic methods is the natural solution, providing near-optimal solutions in a much faster execution time. Choosing which optimization approaches to apply in each time is the focus of this work, considering the requirements for each problem and the information of previous executions. A case-based reasoning methodology is proposed, considering previous cases of execution of different optimization approaches for different problems. A fuzzy logic approach is used to adapt the solutions considering the balance between execution time and quality of results

Keywords: Case Base Reasoning, Optimization Algorithm, Classification

1 Introduction

One of the main objectives of computational intelligence is to impart systems with the ability to reproduce human-like reasoning. Case-based Reasoning (CBR) is an Artificial Intelligence (AI) approach to learning and problem solving based on the past experience, which is usually stored in a case-base (CB) [1]. CBR also captures new knowledge, making it immediately available for solving new problems. AI techniques have excelled in problem-solving as a good solution over conventional techniques.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 641794 (project DREAM-GO) and a grant agreement No 703689 (project ADAPT); and from FEDER Funds through COMPETE program and from National Funds through FCT under the project UID/EEA/00760/2013

CBR has been used in many application domains, one of them being in solving power and energy systems. In [2] a CBR system for building energy prediction is proposed, with the aim at identifying operation issues and proposing better operating strategies. Simplified models based on CBR to predict the hourly electricity consumption of an institutional building are proposed in [3]. A CBR method providing online decision-making for optimization of coal-blend combustion was investigated in [4]. The estimation of the energy performance of new buildings using CBR is studied in [5]. These are relevant contributions that cover some problems in the energy domain. However, many urgently needed issues in this area are still not addressed, such as the energy resource operation and planning.

The Optimal Resource Scheduling (ORS) problem, however, requires extremely heavy computational models, depending on the amount and diversity of the considered resources, and on the depth of network validation and analysis. For this reason deterministic approaches are, most of the times, inadequate [6]. Metaheuristics are proving to be the most suitable alternative, since they are able to reach near-optimal solutions in much faster execution times [7]. These algorithms do not guarantee the optimum global solution, but in turn the response time is much lower compared to the traditional exact algorithms that guarantee it. Many of these methods have also been applied in the resolution of the ORS problem [6, 8].

The question remains, however, on how to make most use of the whole set of available algorithms, depending on the needs and characteristics of each problem. Metaheuristic methods are able to provide approximate solutions in fast execution times, while deterministic approaches need larger times to compute, but are able to provide the optimal solution. Some work has already been made with the application of CBR systems to similar problems, namely in [9], which presents a study to try finding the ideal parameters to apply in evolutionary algorithms. In this work a CBR methodology is used to estimate the best parameter setting for maximizing the performance of evolutionary algorithms. However, in the present work authors propose, not only to adapt the parameterization of a certain algorithm to meet the requirements of execution time versus quality of results, but also to choose the most appropriate algorithm and respective parameterization taking into account the availability of several distinct algorithms of different natures.

This paper thus proposes a CBR based approach that, given the problem characteristics and requirements, and considering an historic CB log of past executions of each algorithm to solve the energy resource optimization problem with different settings, suggests the most appropriate algorithm to apply and the respective parameterization. A problem-driven approach is applied in the retrieve and revise phases, considering the specificities of the different considered variables, and a fuzzy logic based approach [10, 11] is used in the revise phase to adapt the solutions to the requirements of the new problem, namely considering the balance between execution time and quality of results.

After this introductory section, section 2 describes the CBR approach proposed in this paper. Section 3 presents the experimental findings of the application of the proposed approach to a historic CB log of previous executions done by the authors' research team. Finally, section 4 presents the most relevant conclusions of this work.

2 Proposed CBR approach

In this problem, each historic case contains the set of information that is presented in Table 1. The process for which the CBR is oriented refers to choosing the method to use in the problem characterized with different parameters expressed in Table 1. There are 3 types of classification: type A indicates the parameters used for assessing the similarity between case studies, type B indicates the parameters used to determine the quality of each algorithm, and type C are the output parameters. The ID refers to the identification of each case study. The ORS problem contains the type of objective function, where 1 means single-objective optimization problem and 2 corresponds to multi-objective optimization problem. The ORS function parameter refers to which is the ORS problem for the corresponding case study, as can assume 4 states: 1 means minimizing the cost, 2 is minimizing the cost and GHG emissions, 3 is minimizing the cost and demand difference, and 4 is minimizing the cost and voltage deviation.

Table 1. Case structure

Type of parameter		Designation
		ID
A (Similarity)	**A1**	ORS problem
		ORS function
	A2	Period
		Bus
		No. DG quadratic
		EVs
		Congestion power (kW)
B (Quality)		Objective function
		Execution time (s)
C (Decision)		Algorithm
		Parameters

The Period refers to the number of periods of the ORS problem, e.g. 24 hourly periods. The Bus parameter corresponds to the number of buses that compose the distribution network of the case study. This parameter influences the execution time of the algorithms. The No. DG quadratic refers to the number of DG units using the quadratic function for their operation cost. The parameter EVs indicates the number of electrical vehicles used in each case study. The Congestion power refers to the average amount of congestion power of the case study. All these parameters are used by the CBR systems to choose the similar cases. The ORS problem and ORS function parameters have a distinct classification of **A1**, because they are firstly used to filter the cases that were solved for similar ORS problems, i.e. it is mutually exclusive: either a past case is of the same type as the case to be solved or not. On the other hand, all the other Similarity (type **A**) parameters, are classified as **A2**, which means that the similarity between past and current case can be calculated and represented by a value (in this case as a percentage of similarity for each of these parameters).

The type **B** parameters are those that enable determining the quality of the results. The Objective function indicates the objective function result obtained by the algorithm. In the case of multi-objective problems, both objective functions are stored in this parameter. Execution time contains the time that the algorithm took to solve the ORS problem for the corresponding case study. These two parameters are used to select the best algorithms after the CBR approach obtains the similar historic cases by analyzing the type **A** parameters.

Once the quality of the solutions (type **B**) of the similar cases (type **A**) is assessed, a decision is made on which methods and respective parameterizations are the most adequate (using the type **C** parameters). The Algorithm parameter is the name of the algorithm used to solve the case study. Parameters contains the parameters used in each algorithm to solve the historic case, as can be seen in Table 1. These two last parameters are type **C**, because they contain the information on which algorithm and parameters were used to solve the problem. After describing the content of each parameter in the historic cases, the different phases of CBR system is describe in following steps.

2.1 Retrieve

Analyse the ***A1*** parameters for selecting the cases containing the same type of problem (ORS problem) and type of function (ORS function). Each historic case is filtered according to the value of the ORS problem parameter, given by equation (1).

$$F_{HC}^{1} = \begin{cases} HC_j \,, H_j\left(A1_{(i)}\right) = CS\left(A1_{(i)}\right) \\ 0 \,, Otherwise \end{cases} \qquad (1)$$
$$\forall j \in \{1, \dots N_{HC}\}; i = \{ORS\ problem\}$$

Where, F_{HC}^{1} contains the historic cases that were filtered by equation (1). The terms *HC* and *CS* correspond to the historic case and current case study, respectively. The index *j* refers to the *ID* of each historic case, while index *i* corresponds to *ORS problem* parameter. N_{HC} refers to the total number of historic case studies in the database.

Then, the historic cases filtered as (F_{HC}^{1}) are also filtered if they have the same value for the ORS function, by equation (2).

$$F_{HC}^{2} = \begin{cases} F_{HC(j)}^{1}, F_{HC(j)}^{1}\left(A1_{(i)}\right) = CS\left(A1_{(i)}\right) \\ 0 \,, Otherwise \end{cases} \qquad (2)$$
$$\forall j \in \{1, \dots, N_{HC}^{F1}\}; i = \{ORS\ function\}$$

Where, F_{HC}^{2} contains the historic cases that were filtered by equation (2), and index *i* corresponds to *ORS function* parameter. N_{HC}^{F1} corresponds to the total number of historic cases filtered in (1). The historic cases with *ORS problem* equal to 2 (multi-objective problems) that have *ORS function* equal to 2, 3 or 4, i.e. minimizing the cost and other competitive objective, are all considered for a current case study with the same *ORS problem* and containing the same information for the *ORS function* parameter (2, 3 or 4). The idea with this condition is to separate problems with distinct objective function.

Determine the cases that are similar to the current one through the use of *A2* parameters. For each historic case ($F_{HC(j)}^{2}$) the similarity percentage of each *A2* parameter ($P_{HC(j)}^{A2(i)}$) is calculated by equation (3).

$$P_{HC(j)}^{A2(i)} = \begin{cases} \dfrac{F_{HC(j)}^{2}\left(A2_{(i)}\right)}{CS\left(A2_{i}\right)}, CS\left(A2_{(i)}\right) \geq F_{HC(j)}^{2}\left(A2_{(i)}\right) \\ \dfrac{CS\left(A2_{i}\right)}{F_{HC(j)}^{2}\left(A2_{(i)}\right)}, F_{HC(j)}^{2}\left(A2_{(i)}\right) \geq CS\left(A2_{(i)}\right) \end{cases} \qquad (3)$$
$$\forall j \in \{1, \dots, N_{HC}^{F2}\}; \forall i \in \{1, \dots, N_{A2}\}$$

where, N_{HC}^{F2} is equal to the number of historic cases filtered in previous step by equation (2), while N_{A2} corresponds to the total number of ***A2*** parameters. The similarity percentage is calculated by dividing the value of each ***A2*** parameter ($A2_{(i)}$) between the historic and current cases (or *vice versa* - allowing avoiding similarities over than 100%). Then, the average similarity is determined, which corresponds to the similarity percentage of each historic case, and is given by equation (4)

$$PC_{HC(j)} = \frac{1}{N_{A2}} \times \sum_{i=1}^{N_{A2}} P_{HC(j)}^{A2(i)} \quad \forall j \in \{1, \dots, N_{HC}^{F2}\};\ \forall i \in \{1, \dots, N_{A2}\} \tag{4}$$

For a current case with parameters (e.g. *Period*, *Bus* or *EVs*) very close to a historic one, the similarity percentage of each historic case j ($P_{HC(j)}$) will tend to 100%. Finally, filter the historic cases with a similarity percentage ($P_{HC(j)}$) higher or equal to 75%.

$$SC_j = \begin{cases} F_{HC(j)}^{2}, P_{HC(j)} \geq 0.75 \\ 0, Otherwise \end{cases} \quad \forall j \in \{1, \dots, N_{HC}^{F2}\} \tag{5}$$

Where, set SC_j contains all the similar cases.

2.2 Reuse

Extract the algorithms that are used in the similar historic cases and their quality parameters (type ***B*** of Table 1). The same algorithms with different parameters can be considered multiple times, if it is used in multiple similar cases. Steps 3, 4 and 5 are only applied if there is any similar historic case study, otherwise, the CBR systems will select all the algorithms that can solve the chosen ORS problem.

First, filter the algorithms with different parameterization that were used to solve the similar cases, as described in equation (6).

$$Method = SC_j(C_{(i)}) \quad \forall j \in \{1, \dots, N_{HC}^{F2}\}; i = \{Methodology; Parameters\} \tag{6}$$

Where, index i indicates the parameters of type C from Table 1. Second, the average execution time (***B*** parameter) of all cases solved by the same algorithm and parameterization (equation (6)) is determined, because the same algorithm and parameterization might be used by multiple similar cases, which is given by (7).

$$Time_{Met} = \frac{1}{N_{SC}^{Met}} \sum_{j \in SC^{Met}} SC_j(B_{(i)}) \quad \forall\ Met \in \{1, \dots, N_{Met}\}; i = \{Execution\ time\} \tag{7}$$

Where, SC^{Met} refers to the set of all similar cases that were solved by the same algorithm and parameterization with index *Met*. N_{SC}^{Met} contains the number of similar cases solved by the same algorithm and parameterization with index *Met*. N_{Met} refers to the total number of algorithms with different parameterization in (6).

Finally, the average objective function (type B parameter) of all cases solved by the same algorithm and parameterization is also calculated using the previous equation (7). These values are stored in variable Fun_{Met}. Before applying this equation, the objective function values are normalized, because the cases can have objective function values with different magnitudes. The number of considered historical cases is crucial, because with many cases this process can become heavy and slow, so a good historical cases selection (retain phase) is important.

2.3 Revise

Choose the most appropriate algorithms to solve the current case study through the use of a fuzzy method. The variables $Time_{Met}$ and Fun_{Met}, determined in previous step, are used by the fuzzy method. First, create the membership function (μ^{Time}) related to time (efficiency), which is represented in Fig. 1.

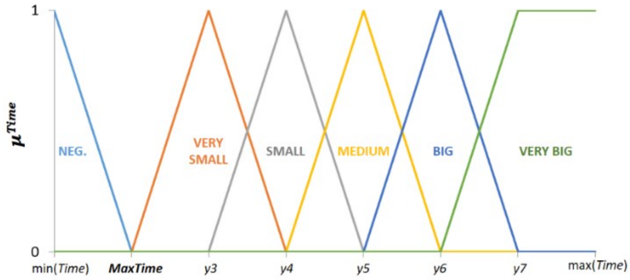


Fig. 1. Membership function of efficiency

The membership function has dynamic intervals to be adapted to every case study. The membership function starts at the minimum *Time* among all methods equation (7) , the second value of this function is the maximum time defined by the VPP in the input data, which is represented as *MaxTime*. The maximum *Time* occupies the other extreme of the membership function. The remaining values ($y3$, $y4$, $y5$, $y6$ and $y7$) are proportionally distributed between the *MaxTime* and the maximum time. The $Time_{Met}$ equation (7) of each method *Met* is classified based on this membership function, which indicates how much far the *Time* is from the *MaxTime* (i.e. NEGATIVE, VERY SMALL, SMALL, MEDIUM, BIG or VERY BIG).

Secondly, the membership function (μ^{Fun}) related with objective function (effectiveness), is created, which is represented in Fig. 2.

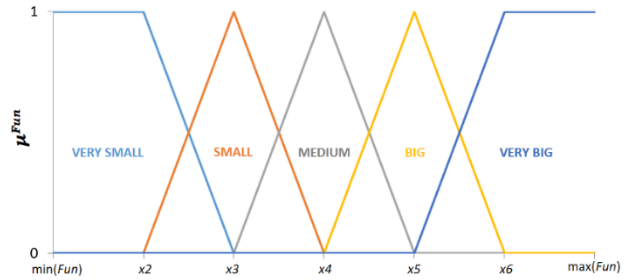


Fig. 2. Membership function of effectiveness

This membership function also has dynamic intervals, as it starts with the minimum *Fun* among all methods, while the maximum *Fun* is placed in the other extreme of the function. Just like the previous one, the remaining values are proportionally distributed between the minimum and maximum *Fun*. The Fun_{Met} is be classified based on this membership function, which also indicates how far the Fun_{Met} of each method *Met* is from the minimum *Fun*.

Then, select the algorithms considering the μ^{Time} and μ^{Fun} classifications by equation (8).

$$Method = \begin{cases} Method_{Met}, \mu_{Met}^{Time} = NEGATIVE \\ Method_{Met}, \begin{matrix} \mu_{Met}^{Time} = \{VERY\ SMALL; SMALL\} \\ \mu_{Met}^{Fun} = \{VERY\ SMALL; SMALL\} \end{matrix} \end{cases} \quad (8)$$
$$\forall\ Met\ \in \{1, \dots, N_{Met}\}$$

The methods with μ^{Time} equal to NEGATIVE, which means an execution time below the *MaxTime*, are accepted to solve the current case, without considering their effectiveness classification (μ^{Fun}). The other methods with a time slightly higher than *MaxTime*, which have VERY SMALL and SMALL efficiency classification (μ^{Time}), are accepted if they also have an objective function close to the minimum, which are VERY SMALL and SMALL classifications for the effectiveness membership function (μ^{Fun}). All methods that are classified as bigger are automatically excluded, since their execution time is too big to useful for the considered problem or the results quality is too low (big difference from the best methods).

Finally, the fuzzy confusion matrix, which joins the two membership functions (μ^{Time} and μ^{Fun}), is applied to take actions regarding the methods with VERY SMALL and SMALL classifications. Basically, these methods are changed in terms of their parameterization to reach a lower execution. The amount of these changes will be given by the fuzzy confusion matrix, which can be consulted in the fuzzy confusion matrix presented in Table 2. This enables to consider methods that would be excluded because they are above *MaxTime*, but have good objective function results.

Table 2. Fuzzy confusion matrix for small and very small classifications of effectiveness and efficiency

Efficiency Classification	Effectiveness Classification	Action to take
VERY SMALL	VERY SMALL	Very small reduction
	SMALL	Small reduction
SMALL	VERY SMALL	Small reduction
	SMALL	Big reduction

2.4 Retain

Evaluate the possibility of storing the results of the current case study in the database of historic cases. Determine the similarity of the current case study (P_{CS}) by applying the equations (4) and (5), include the current case in the database of historic cases, if its similarity percentage is lower or equal to 95%, defined as equation (9).

$$HC = \begin{cases} CS, P_{CS} \leq 0.95 \\ 0, Otherwise \end{cases} \quad (9)$$

A current case with a percentage higher than 95% is not adding new value to the historic cases, since it is only bringing useless information to the processed.

3 Results

This section presents the experimental findings concerning the application of the proposed methodology to a new case. 21 previous cases are considered in the CB, which refer to different executions of several algorithms with different parameterizations, to different variations of the ORS problem. The new case is defined by the next conditions: {ID=_; ORS problem=1; ORS function=1; Period=24; Bus=37; No DG quadratic= 3; EV's=2000 and Congestion power= 730}, B and C parameters present in Table 1 will be find by CBR methodology.

To carry out the CBR process, the new case must contain all elements of group **A** (Similarity). Table 3 shows the results of the different methods selected by the equations corresponding to the group of similarities. Please refer to [8] for a detailed description of the optimization methods shown in the last column of Table 3.

The results of equations (1), (2), (4) and (5) are related to the retrieve process, and equation (6) is already the initial phase of the reuse process, where similar cases are identified. As can be seen, the cases filtered by the ORS problem and ORS function are the same (20 cases). By applying the calculation of the total similarity (equation (5)) 4 cases are excluded, being 16 cases considered similar to the new case.

Table 3. Results similarity

Equation (1) - case ID	Equation (2) - case ID	Equation (4)	Equation (5)	Equation (6)
1	1	0,4842822	✗	-
2	2	0,2817593	✗	-
4	4	0,9922183	✓	RSA
5	5	0,9953471	✓	HSA
6	6	0,9956742	✓	ERS^{2}A
7	7	0,995853	✓	PERS^{2}A
8	8	0,9958794	✓	SADT
9	9	0,9942107	✓	GA
10	10	0,9928798	✓	PSO
11	11	0,9953506	✓	PERSGA
12	12	0,9953364	✓	PERSPSO
13	13	0,9956334	✓	GADT
14	14	0,995663	✓	PSODT
15	15	0,9960788	✓	MINLP
16	16	0,9613657	✓	PERS^{2}A
17	17	0,9615992	✓	SADT
18	18	NaN	✗	-
19	19	0,9988149	✓	PERS^{2}A
20	20	0,9998657	✓	SADT
21	21	NaN	✗	-

Table 4 presents the efficiency classification, as result of the efficiency fuzzy variable, and the effectiveness classification of the application of each of the selected methods to the selected cases. In Table 4, the 16 similar cases are filtered by method, and may have different configurations within the same method. In this case, the average of these configurations (execution time and objective function) is made. In Table 4, 12 methods are present which means that there are repeated methods. Being that PERS^{2}A and SADT repeated three times. The values are sorted by execution time value, in an ascending order. The fuzzy results related to the value of the objective function, i.e. the effectiveness of each method and respective parametrization in solving the previous problem identified as similar to the new case. Table 4 also presents the decision results, which are a direct output from the confusion matrix that combines the fuzzy results for efficiency and effectiveness of each method.

Table 4. Efficiency and effectiveness results

Method	Equation (7) Time (s)	Equation (7) Objective function	Confusion Matrix		Type of modification
			Efficiency	Effectiveness	
ERS^{2}A	54,1	23944,94	NEG.*		-
RSA	174,28	24375,45	NEG.*		-
PERS^{2}A	189,43	25415,76	NEG.*		-
SADT	393,4367	25446,97	NEG.*		-
PERSPSO	482,88	23986,35	VERY SMALL	VERY SMALL	Very small reduction
PSODT	544,11	23946,32	VERY SMALL	VERY SMALL	Very small reduction
PSO	550,91	24291,86	VERY SMALL	SMALL	Small reduction
HSA	598,35	23985,04	VERY SMALL	VERY SMALL	Very small reduction
PERSGA	635,67	23984,61	VERY SMALL	VERY SMALL	Very small reduction
GADT	673,47	23949,94	VERY SMALL	VERY SMALL	Very small reduction
GA	1731,54	24125,38	-	-	Excluded
MINLP	94941,85	23895,53	-	-	Excluded

*Equation (8)

In Table 4 are expressed the decision results obtained by the CBR system. As it can be seen, if the classification in the efficiency process is Negative, the method will be accepted without any change. On the other hand, if the classification is any other, the value of the objective function is analyzed, the classifications medium, big and very big, are excluded at the beginning. The confusion matrix is only executed for the methods classified as very small and small. The result of the confusion matrix gives the type of modification that is required to execute so that the given method can obtain an execution time value lower than the one defined as *MaxTime* by 400 seconds.

By applying the rules of the fuzzy processes, the possible methods to solve the problems went from 12 to 10, and the MINLP and GA were excluded. The ERS2A, RSA, PERS2A and SADT methods were accepted without any change. The remaining methods are subjected to a certain type of change to be performed, which regards the adaptation of the method's parameterization, e.g. using a smaller number of iterations or a smaller number of particles in the PSO to achieve faster results.

4 Conclusions

This paper presented a CBR methodology to support the choice of the methods to use in solving the energy ORS problem. The proposed method includes a fuzzy based process to determine the changes in parameterization that should be applied to each method that is considered promising to solve a new case with specific characteristics.

It is clear that this method brings advantages when compared to a manual process, because choosing manually hardens the effectiveness of the choice, and the time spent, e.g. in the choice of parameters.

The performance of CBR systems is highly correlated with the number of cases that it imbues. Even so, the presented results suggest as final result a considerable number of methods to solve the problem, all of which with expected small execution times and good quality of results for the envisaged problem. This means that the presented methodology was effective in the selection and classification of the methods. The modifications to be performed in the methods, as result from the fuzzy process, enlarge the scope of possible methods to be applied, as rather than excluding such methods for being just a bit slower or presenting a bit worst quality of results than other methods, it still considers the most promising ones as possible solutions, subject to a degree of changes that would make them suitable to solve the problem as well.

As future work, it is intended to develop a method for deciding which parameters to modify to obtain the given value of maximum execution time, according to the results of the fuzzy process. It is also proposed to apply decision trees in the process of retrieve. Finally, the process of reviewing can be enhanced with the help of an expert, in order to build an expert system to perform the revision of the changed parameters.

References

1. Aamodt A, Plaza E (1994) Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun IOS Press 7:39–59.
2. Monfet D, Corsi M, Choinière D, Arkhipova E (2014) Development of an energy prediction tool for commercial buildings using case-based reasoning. Energy Build 81:152–160. doi: 10.1016/j.enbuild.2014.06.017
3. Platon R, Dehkordi VR, Martel J (2015) Hourly prediction of a building's electricity consumption using case-based reasoning, artificial neural networks and principal component analysis. Energy Build 92:10–18. doi: 10.1016/j.enbuild.2015.01.047
4. Xia J, Chen G, Tan P, Zhang C (2014) An online case-based reasoning system for coal blends combustion optimization of thermal power plant. Int J Electr Power Energy Syst 62:299–311. doi: 10.1016/j.ijepes.2014.04.036
5. Hong T, Koo C, Kim D, et al (2015) An estimation methodology for the dynamic operational rating of a new residential building using the advanced case-based reasoning and stochastic approaches. Appl Energy 150:308–322. doi: 10.1016/j.apenergy.2015.04.036
6. Soares J, Sousa T, Morais H, et al (2013) Application-Specific Modified Particle Swarm Optimization for energy resource scheduling considering vehicle-to-grid. Appl Soft Comput 13:4264–4280. doi: 10.1016/j.asoc.2013.07.003
7. Baños R, Manzano-Agugliaro F, Montoya FG, et al (2011) Optimization methods applied to renewable and sustainable energy: A review. Renew Sustain Energy Rev 15:1753–1766. doi: 10.1016/j.rser.2010.12.008
8. Sousa T, Morais H, Castro R, Vale Z (2016) Evaluation of different initial solution algorithms to be used in the heuristics optimization to solve the energy resource scheduling in smart grids. Appl Soft Comput 48:491–506. doi: 10.1016/j.asoc.2016.07.028
9. Yeguas E, Luzón MV, Pavón R, et al (2014) Automatic parameter tuning for Evolutionary Algorithms using a Bayesian Case-Based Reasoning system. Appl Soft Comput 18:185–195. doi: 10.1016/j.asoc.2014.01.032
10. Faia R, Pinto T, Vale Z (2015) Dynamic Fuzzy Estimation of Contracts Historic Information Using an Automatic Clustering Methodology. In: Bajo J, Hallenborg K, Pawlewski P, et al (eds) Highlights PAAMS Collect. SE - 23. Springer International Publishing, pp 270–282
11. Chang P-C, Fan C-Y, Dzan W-Y (2010) A CBR-based fuzzy decision tree approach for database classification. Expert Syst Appl 37:214–225. doi: 10.1016/j.eswa.2009.04.062

A workflow cloud management framework with process-oriented case-based reasoning

Eric Kübler and Mirjam Minor

Goethe University, Business Information Systems, Robert-Mayer-Str. 10, 60629 Frankfurt, Germany
ekuebler@cs.uni-frankfurt.de,
minor@cs.uni-frankfurt.de

Abstract. Workflow execution in the cloud is a novel field for process-oriented case-based reasoning (PO-CBR). In contrast to traditional PO-CBR approaches where the focus is usually on single workflow instances, an entire set of workflow instances is considered that are currently running on cloud resources. While traditional methods such as running a workflow management tool monolithically on cloud resources lead to over- and under-provisioning problems, other concepts include a very deep integration, where the options for changing the involved workflow management tools and clouds are very limited. In this work, we present the architecture of *WFCF*, a connector-based integration framework for workflow management tools and clouds to optimize the resource utilization of cloud resources for workflow by Case-Based Reasoning. Experience reuse contributes to an optimized resource provisioning based on solutions for past resource provisioning problems. The approach is illustrated by a real sample workflow from the music mastering domain.

1 INTRODUCTION

Resource provisioning for workflow execution is a well known issue in workflow management. It has been solved for on-premise systems by load balancing components, for instance. However, in cloud computing, resources are provided on-demand. Thus, workflow management in the cloud has to deal with scalable resources. Standard load balancing approaches are not capable to deal with this. Novel business concepts for workflow execution in the cloud emerge. One of these concepts is *workflow as a Service (WFaaS)* as introduced by [17, 9]. The Workflow Management Coalition [18] defines a *workflow* as "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules". A *task*, also called activity, is defined as "a description of a piece of work that forms one logical step within a process. An activity may be a manual activity, which does not support computer automation, or a workflow (automated) activity. A workflow activity requires human and/or machine resources(s) to support process execution" [18]. The idea of WFaaS is to execute activities within a cloud. A cloud vendor [3] is a company that offers services in the cloud, for example the execution of a workflow. However, the vendor is not always a cloud provider. Even if renting the required cloud resources by a third party provider, the vendor is responsible for maintaining the service level agreements (SLA) for the own costumers. An SLA defines agreements between the provider and the customer about different aspects of the quality of service. For example, an SLA can be specified for the execution time of the workflow. To prevent an SLA violation, the vendor may rent more resources than required (over-provisioning) but this will reduce the profit. On the other hand, if the vendor rents less resources than required (under-provisioning) this can lead to violations of the SLA. Violations of an SLA create high costs and a loss of reputation [16]. Thus, the optimal management of resources is an important aspect for cloud computing [6] in general and, particularly, for WFaaS vendors. It is challenging to find a good balance between over- and under-provisioning of resources [4]. A straight-forward solution to provide resources is the static way. This means, the system does not adjust itself to a changing situation. Obviously, this will lead to under- or over-provisioning [16]. A more dynamic approach is preferable. Existing approaches range from rather simple, rule-based solutions, such as observing the number of open connections to a cloud resource [14] to sophisticated, algorithmic solutions [15].

Knowledge and experience management methods [7] provide an alternative solution approach focusing on the reuse of experience. In this paper, we investigate *Case-based reasoning (CBR)* as a method for optimizing the provisioning of cloud resources by experience reuse. This work is an extended version of the approach we introduced in ICCBR 2014 [13]. We will introduce the architecture of *WFCF* (Workflow Cloud Framework) a connector-based integration framework for workflow management tools and clouds that aims to optimize the resource utilization of cloud resources for workflows by means of CBR. WFCF follows a shallow integration approach, i.e. it is independent of the chosen workflow management tools and cloud systems. The idea is to have a set of WFCF components that are independent of the workflow management tools and cloud systems. A tool-specific set of connectors interacts with the actually used tools and system. Further, we will present the details how the problem solving component of the architecture is realized by means of PO-CBR. The benefits of using PO-CBR is twofold namely the reduction of costs for the vendor by reducing over-provisioning and SLA violations and, second, a better cost estimation based on experience.

2 WFCF ARCHITECTURE

In this section, we will explain the architecture of WFCF and its components. Starting with the overall architecture, we show the details of the monitoring and management components and how they interact.

2.1 Overall architecture

Figure 1 shows the overall architecture of the WFCF, which we will explain in the following. The architecture can be divided roughly in three parts: the environment, the monitoring component and the management component. The environment are the cloud and the workflow management tool that is used by the customer. Ideally, WFCF will use the already offered information and management methods of the tools, so that additional changes are not necessary. Therefore, WFCF will use offered log files, databases and API's for monitoring the environment and to configure the cloud. *CWorkload* is the monitoring component. It collects information from the environment and combines data across the different layers (the cloud layer and the workflow layer) to one status model of the system. We had done initial tests for the cross layer monitoring aspect of CWorkload in [10]. The management component recognizes current or upcoming problems within the system. This could be for example violated SLA's, violated constraints or resource over-provisioning. If a problem occurs, the management component searches for a solution and reconfigures the cloud. We will explain this in more detail in section 2.3.

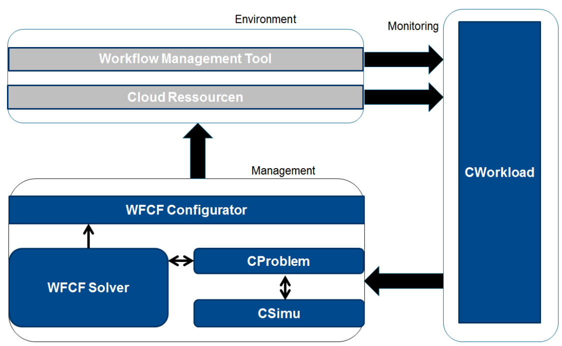


Fig. 1. Architecture of WFCF

2.2 Monitoring

The main components of WFCF work independently from the actually used environment. To work properly, WFCF needs information about the status of the actually running workflow instances and the resource utilization of the cloud. The basic idea is, to connect the used workflow engine and cloud to WFCF with connectors. This allow the usage of different engines and clouds without or just small adaptions. For further details, please have a look at our previous work [12].

2.3 Management

Whereas the monitoring component observes the environment, the management component configures it. This means, the management component starts and stops virtual machines or PaaS container, scales resources and migrates content. Figure 2shows the management component in more detail. After CWorkload has built the WFCF CloudWF Status, *CProblem* is the part of WFCF which interprets the current status of the environment that is recorded as the WFCF CloudWF Status. Besides the CloudWF status, there is another archive, the *Global SLA // Constraint Archive*, where global constraints and SLA's are stored. The *Global SLA // Constraint Archive* contains SLA's and constraints that are valid for all workflows of a user. There are several different problems that can occur and which CProblem will identify, e.g., violated SLA's. We are planning that CProblem does not only check the current situation, but also do a forecast to identify upcoming problems and over-provisioning. A workflow definition contains all information about the structure of the workflow. For example, the name of the tasks and their order. Via the workflow definitions, for example, CProblem can recognize if a certain web service is going to be used in the future by a currently running workflow instance. If not, WFCF can shut down the VM or container to save money. Another possible scenario could be that currently, there is no violated SLA, but in the near future, several tasks with high resource demand will be started, which can probably lead to a SLA violation, so WFCF should scale up the resources to avoid this problem. Forecasting SLA violations, however, could be a difficult task. To decide if the start of some resource intensive tasks lead to a SLA violation is not as easy as to recognize if a web service has not started yet. A simulations seems a proper way to identify these kind of problems. Therefore, CProblem interacts with *CSimu*. We are planning to use CloudSim [1] as the core of our simulation part. CSimu will simulate the execution of the tasks with the current cloud status and will show if this will lead to a SLA violation. If any problem is unidentified, CProblem extends the CloudWF status with annotations about the problems. This new annotated model is the *WFCF CloudWF Problem*. Such annotations could be, for example, *web service x is not longer needed* or *SLA y is currently violated*. Whereas CWorkload is the core of the monitoring component, the *WFCFSolver* is the core of the management. Similar to CWorkload, the solver has two jobs. First, the solver searches for a new cloud configuration that solves the current problems. Then it finds a reconfiguration path from the current cloud configuration to the new solution. In the last step, the solver sends the reconfiguration steps to the *WFCF Configurator* as shown in Figure 1. The reconfigurator then will do the reconfiguration job. There are several possible approaches to find a new cloud configuration. We will choose *Case-Based Reasoning* (CBR) as our solving strategy.

3 CBR FOR PROBLEM SOLVING

In this section we take a closer look how the *WFCFSolver* will solve the cloud management problems with CBR methods. As mentioned in Section 1, the idea

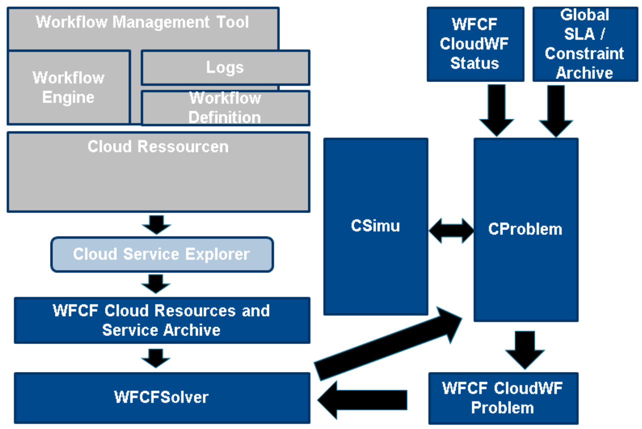


Fig. 2. Management of WFCF

of CBR is that similar problems have similar solutions. If a problem situation occurs the system retrieves experience by searching a similar situation from the past. In our case a problem situation is a cloud configuration with a problem, such as violated SLA's. This is the retrieval step. The key to experience retrieval is a good notion when some kind of experience is relevant for a certain situation. This knowledge is captured in the similarity measure [7]. The reuse step of CBR is to use the solutions from the past for the current problem. In our case, the solution contains re-configuration steps. This for example could be the to start new VM's or to migrate containers to another VM.

A problem situation is recorded as *WFCF CloudWF Problem*. Figure 3 shows an example of a simple CloudWF Problem. This example contains one VM, two containers for the required web services and a bunch of workflow instances currently being executed. The image depicts not the entire workflows but the tasks that are currently active within the instances. Most of the workflow instances are derived from the same workflow definition and are in the same state of execution. At this point, the task *Task 1* uses the web service *web service 1* while *Task 2* uses *web service 2*. In addition, there is another workflow instance (in the bottom right corner). This instance is probably from a different workflow definition, or the instance is in a different state of execution. The current task of this instance is *task 217* and for its proper execution, a web service that has not yet started is required. This example also includes the constraint that the average resource utilization must not extend 75% for reasons of performance. The example CloudWF Problem includes also three problems. The resource utilization of the CPU and memory of VM1 is too high and a new web service must start for Task 217. More complex CloudWF Problems may involve several VM's, containers and workflow instances.

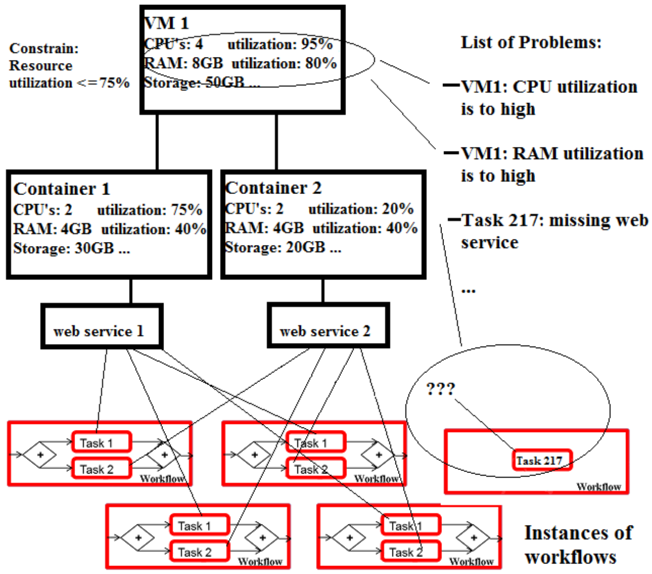


Fig. 3. Example representation of a case

A *case base* is an archive of previous problems and their solutions. The case base is not depicted in Figure 2, because it is part of the solving strategy and not part of WFCF itself. The solver will search the case base for similar problems in the past. In our previous work [11], we have introduced the idea of a similarity function for cloud configurations. For the similarity of a cloud configuration, we consider the following aspects as important.

The provided resources. Two VM's are similar, if they have a similar set of resources available. For example, two VM's with a quad core processor should be more similar than a VM with a dual core processor and a VM with a quad core. The idea is, that VM's with a similar set of resources should handle general workload similar, where VM's with a different set of resources maybe lead to other results, for example you can not migrate a container that requires a quad core, if the VM only have a dual core. The same applies to containers.

The resource utilization. VM's with a similar resource utilization, for example average CPU usage, should be considered as similar. If the utilization differs significantly, a solution that is valid for one case could be invalid for the recent case. For example if the disk space utilization for a VM $vm_1$is 20% and for an another vm_2 100%, the system can not migrate a container to vm_2, because of the lack of free disk space, while a migration to vm_1 is feasible. The same applies to containers.

The assigned SLA's and whether they are violated or not. If two cloud configurations have a similar set of SLA's, the configurations should be considered as similar. Different SLA's or the violation of different SLA's can lead to a situation, where a problem of the one case is not a problem in another case. For example if a cloud configuration includes an SLA on availability and the other doesn't, the availability can be a problem in the first case while it is not in the second. That leads to the situation, that a solution that mends the availability problem for one case is not applicable for the other case.

The executed workflow instances and their workflow definitions. The number of the started instances and the structure of the workflow definitions can have a high impact on the requirements for resources and for started web services. For example, if an instance of a workflow is started that requires a certain web service, every solution that does not include this web service is not valid. The structure of the workflow definitions also specifies which tasks will be started next.

To determine the similarity of two cases, we use a composite, distance-based similarity function based on the aspects introduced before. The similarity of each aspect in two cases is computed by a particular local similarity function. The local similarity values are aggregated by means of a sum of weighted aspects. For example, the similarity function of the resources provided for a VM is based on a taxonomy, and analog for containers. For the size of the provided resources, we have been inspired by Amazon EC2 instances [5] for nodes and OpenShift [2] for containers. For other aspects, we use mainly standard distance functions. For example to determine the distance between the resource utilization for VMs vm_uti, we use the Euclidean distance for the resource vectors of CPU, memory, storage, network traffic, and so on. The utilization values are provided in percentage. The distance of the resource utilization vm_util is calculated by the Euclidean distance $vm_{util}(p,q) = \sqrt{\sum_{i=1}^{n}(q_i^{vm} - p_i^{vm})^2}$, where p is the vector of n utilization values for the first case and q for the second case. For example, $q_1^{vm} = 50$ is the utilization of the CPU q_1^{vm} with a value of 50%. p_2 is the utilization of the memory and so on.

The similarity function for the workflow aspect of our approach is ongoing work. Each workflow instance has 0 to n active tasks. These are the tasks that are currently executed. We are planning to consider the currently active tasks, as a *bag of tasks* in our similarity function. In addition, another relevant set of tasks can be derived from the workflow definition namely the set of tasks that will be active in the near future. We call this the *bag of tasks approaching next*. We assume, these two bags of task should be an important part within the similarity function. The similarity of two individual tasks will be determined by its service characterization, the size of its input data and the name of the task. Two tasks are similar if they have the same characterization (for example CPU intensive) and if the size of the input data and the name of the task are similar. However, we have not decided yet how to implement the similarity function for bags of tasks finally.

For the reuse step, a solution is a cloud configuration without problems. The solver will search for a similar problem and use the solution for this old problem or the solution can serve as a starting point for a new solution. Anyways, the solver will send the solution back to CProblem to check if the solution comes up with new problems. CProblem will check and simulate the solution and give feedback to the solver. This will be repeated until a solution is found or another condition is fulfilled. This could be, for example, a time limit. In this case, the solution with the least significant problem will be chosen. The usage of CBR also opens the opportunity for post-mortem analysis and improvement of the stored solution, while WFCF is otherwise idle. This lazy learning can also be used, if there is no similar soluation, or when the case base is empty. In such a case, a simple rule based approach can generate a first placement for the current situation and a post-mortem analysis can improve the result afterward, for the next time, a similar situation approaches. In addition to the case base, there is the *WFCF Cloud Resources and Service Archive*. This archive contains information about the available type of containers, VM's, web services and so on. This archive helps the solver to find valid solutions. Similar to the connectors in the monitoring part, the *Cloud Service Explorer* is a connector to the cloud to discover available sizes and services and store them in the Resources and Service Archive.

4 EXAMPLE

To demonstrate the idea of WFCF, we will give a running example. As our example domain, we chose music workflows to mastering music. The purpose of such a workflow is to transform and process a music file. This includes to normalize and limit the volume of the sound, increase or reduce the sample rate, convert from mono to stereo or reverse and adding special effects like fading and compressing the size of the music file. Figure 4 shows an example workflow. The workflow is modelled in BPMN [8]. To simplify the image, figure 4 does not show the input and output files of the web services. The workflow starts with the *Init Workflow Parameter* tasks to initialize the workflow by a human. The user chooses some parameter for the later mastering. The following two tasks are also human tasks require along with the first one no cloud resources. The following tasks are all based on web services and alter the music file each time. For example, the task *normalize* normalizes the volume of the music file, while the task *fading* adds a fade-out effect to the end of the music. Let us assume that *task choose file* is currently active.

CProblem realizes that, in the near future, the task *normalize* will start. This task requires the web service *normalize web service* that is not available at the moment and this is a problem. *CProblem* prepares the *WFCF CloudWF Problem* and annotates that this web service is required. Because of the simple cloud configuration and because no SLA's are involved, no simulation from *CSimu* is needed. The *WFCFSolver* searches its case base for a case where a web service is required and no container is currently started. Let us assume that the WFCF-Solver finds such a solution and this solution includes to start a container with the needed web service. The solver will send this solution back to *CProblem* to

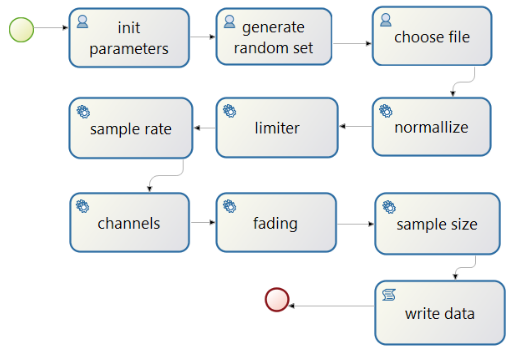


Fig. 4. Sample workflow of mastering music

check if the solution includes new problems. This, however, is not the case. The solver can now start to plan the reconfiguration. After the solver is done, the *WFCF Configurator* starts a container with the web service.

5 CONCLUSION

In this paper, we introduced the architecture of *WFCF*, a connector-based integration framework for workflow management tools and clouds. The goal of WFCF is to provide a way to integrate different workflow management tools and clouds, while also optimizing the resource utilization of the used cloud resources by PO-CBR. To achieve this goal, WFCF uses multiple concepts. The connector's concept allows in a modular way to integrate workflow tools and clouds by using their usual management and monitoring concepts and without the need for special requirements to the used tools. The monitoring component of WFCF analyzes the run time behavior and resource usage of tasks for a better understanding of their needs and also combines information of the workflow management tool and the cloud to a status model for future analysis and forecast of problems. The management component analyzes this status model for problems by using a combination of simulation and static methods. When a problem occurred or can be forecasted, the management component uses CBR to find a similar problem in the past and solve the problem based on the past solution. WFCF aims at a shallow integration of cloud and workflow management tools for flexible combination of tools and the optimization of resource usage. We believe that the use of PO-CBR will lead to the reduction of costs for the vendor by reducing over-provisioning and SLA violations and, second, offer the opportunity for a better cost estimation due to experience, while the approach should be less compute intensive and therefore faster as other solutions. Currently, we are working on a prototypical implementation of the of the architecture to evaluate the concept in future. For our future evaluation, we are planing to compare

WFCF with Cloud Socket. An open issue is to design the similarity functions in detail and the *WFCF CloudWF Status* model in a universal way without dependencies of the actually used tools. Another future task is the acquisition of a larger set of problems that should be recognized and solved and also to investigate how strong is the impact of different optimization goals (for example, reduce costs or reduce SLA violations), for different solutions.

References

1. The CLOUDS lab: Flagship projects - gridbus and cloudbus (2016), http://www.cloudbus.org, 2016-12-08
2. OpenShift (2016), https://www.openshift.com/, 2016-12-08
3. Antonopoulos, N., Gillam, L.: Cloud computing. Springer (2010)
4. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing 53(4), 50–58 (2010)
5. AWS: Amazon web services (AWS) - cloud computing services (2016), http://aws.amazon.com/, 2016-12-08
6. Baun, C., Kunze, M., Nimis, J., Tai, S.: Cloud Computing - Web-Based Dynamic IT Services. Springer (2011)
7. Bergmann, R.: Experience management: Foundations, development methodology, and Internet-based applications. Springer Verlag (2002)
8. Chinosi, M., Trombetta, A.: BPMN: An introduction to the standard 34(1), 124–134 (2012)
9. Korambath, P., Wang, J., Kumar, A., Hochstein, L., Schott, B., Graybill, R., Baldea, M., Davis, J.: Deploying kepler workflows as services on a cloud infrastructure for smart manufacturing 29, 2254–2259 (2014)
10. Kübler, E., Minor, M.: Towards cross-layer monitoring of cloud workflows. In: Helfert, M., Ferguson, D., Muoz, V.M. (eds.) CLOSER 2015, Lisbon, Portugal, 20-22 May, 2015. pp. 389–396. SciTePress (2015)
11. Kübler, E., Minor, M.: Towards a case-based reasoning approach for cloud provisioning. In: CLOSER 2016, Rome, Italy 23-25 April, 2016. vol. 2, pp. 290–295. SciTePress (2016)
12. Kübler, E., Minor, M.: WFCF - a workflow cloud framework. In: CLOSER 2017, Porto, Portugal 24-26 April, 2017. pp. 518–523. SciTePress (2017)
13. Minor, M., Schulte-Zurhausen, E.: Towards process-oriented cloud management with case-based reasoning. In: Proc. ICCBR 2014. pp. 303 – 312. LNCS 8766, Springer (2014)
14. Pousty, S., Miller, K.: Getting Started with OpenShift. "O'Reilly Media, Inc." (2014)
15. Quiroz, A., Kim, H., Parashar, M., Gnanasambandam, N., Sharma, N.: Towards autonomic workload provisioning for enterprise grids and clouds. In: Grid Computing, 2009 10th IEEE/ACM International Conference on. pp. 50–57. IEEE (2009)
16. Shoaib, Y., Das, O.: Performance-oriented cloud provisioning: Taxonomy and survey abs/1411.5077 (2014)
17. Wang, J., Korambath, P., Altintas, I., Davis, J., Crawl, D.: Workflow as a service in the cloud: Architecture and scheduling algorithms 29, 546–556 (2014)
18. Workflow Management Coalition: Workflow management coalition glossary & terminology (1999), http://www.wfmc.org/resources 2016-12-15

Project EVER: Extraction and Processing of Procedural Experience Knowledge in Workflows

Ralph Bergmann[1], Mirjam Minor[2], Gilbert Müller[1], Pol Schumacher[3]

[1]University of Trier, Business Information Systems II
54286 Trier, Germany
[bergmann][muellerg]@uni-trier.de
http://www.wi2.uni-trier.de

[2]Goethe University, Department of Informatics
60629 Frankfurt/Main, Germany
minor@informatik.uni-frankfurt.de
http://www.wi.informatik.uni-frankfurt.de

[3]Celonis SE
81373 München, Germany
p.schumacher@celonis.com

Abstract. The goal of the EVER project (Extraction and Processing of Procedural Experience Knowledge in Workflows), funded by the German Research Foundation, is to investigate new methods in Process-Oriented Case-Based Reasoning and related fields for extracting, representing, and processing procedural experiential knowledge in Internet communities. This paper summarizes the main achievements of the first funding period of this project. The main research addressed the extraction of workflows from textual sources in Internet Communities, the similarity-based retrieval of workflows for a particular goal of a user, and the automatic adaptation of retrieved workflows.

1 Introduction

Today's Social Web allows people in a community of practice to post their own experiences in a diversity of content repositories such as blogs, forums, or Q&A websites [22]. However, today there is no automated support for reusing these rich collections of personal experience. Current search functions available merely consider experience as text to be indexed as any other text and searched as any other document. The objective of the EVER project (Extraction and Processing of Procedural Experience Knowledge in Workflows) is the analysis, the development, and the experimental application and evaluation of new knowledge-based methods, particularly from process-oriented case-based reasoning (POCBR) [7,8,13], information extraction, and machine learning.

The EVER project is funded by the German Research Foundation (DFG) and led by the Universities of Trier and Frankfurt. During the first funding period from 2011 – 2016, the project focused on the reuse of procedural experiences published by private people in Internet Communities such as cooking web sites. In this regard, it was investigated whether workflow technology and POCBR can help to analyze and reuse procedural experiential knowledge from these Internet communities. In the course of this project, several significant contributions to POCBR research have been made, particular in the fields of workflow extraction from text, workflow retrieval, and workflow adaptation. The methods have been consistently evaluated in the domain of cooking recipes. This paper presents a summary of those achievements and shows, how they are connected to draw an overall picture of POCBR.

2 Architecture for POCBR

The overall architecture of our POCBR approach in the EVER project is illustrated in Fig. 1. First, procedural experience is gathered from Internet communities (or alternatively from repositories of workflows in the Business Process Model and Notation (BPMN) format) and stored in a suitable representation. More precisely, a case base of semantic workflows is constructed by extracting workflows from textual sources. The workflows in this repository can be reused, i.e., for a particular problem situation a suitable process represented as workflow can be suggested. This is primarily achieved by retrieving the best matching workflow from the repository. If required, the workflow is automatically adapted according to the requirements and restriction in the particular scenario. The required adaptation knowledge is automatically learned from the case base. In addition to these steps (which basically correspond to the phases of the R^4-CBR cycle [1]) we also include specific methods for user interaction, enabling a conversational POCBR approach [30]. In the following section, we will summarize our research related to various components of the architecture.

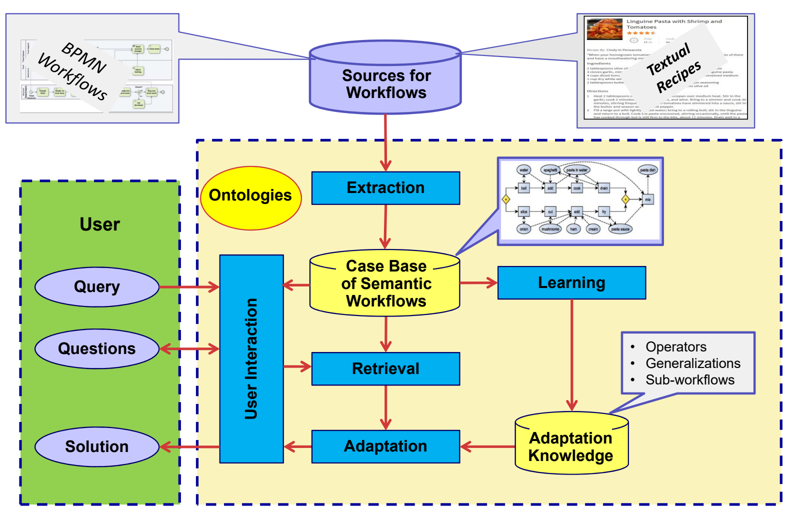


Fig. 1. EVER architecture for POCBR.

3 Semantic Workflows as Case Representation

In order to formalize procedural experience, we employed *semantic workflows* as case representation. Broadly speaking, a workflow consists of a set of *activities* (also called *tasks*) combined with *control-flow structures* like sequences, parallel (AND) or alternative (XOR) branches, as well as repeated execution (LOOP). In addition, tasks consume and produce certain *data items*, or objects, depending on the workflow domain (e.g., ingredients in the cooking domain). Tasks, data items, and relationships between the two form the *dataflow*. For the given application domain, a cooking workflow describes the preparation steps required and ingredients used in order to prepare a particular dish. Here, the tasks represent the cooking steps and the data items refer to the ingredients being processed by the cooking steps. An example cooking workflow for a sandwich recipe is illustrated in Fig. 2.

As a basis for the project, we developed a graph-based representation of semantic workflows that further enables to compute similarities between two workflows [2]. In a semantic workflow the individual workflow elements are annotated with ontological information. In particular, tasks and data nodes are linked into domain-specific task and data ontology and can be further specified by properties, e.g. to represent context factors or resources. In the cooking domain a taxonomy of cooking ingredients and cooking steps is consequently constructed. Within the developed POCBR system CAKE ontologies are represented in an object-oriented fashion while a (partial) transformation into OWL has been developed.

4 Automatic Workflow Extraction from Text

Prior to reasoning with procedural knowledge, the available experience is transformed into a suitable and formal process representation. More precisely, we developed a novel framework for automated workflow extraction [23], which transforms textual descriptions of processes into semantic workflows. Here, from the textual description the

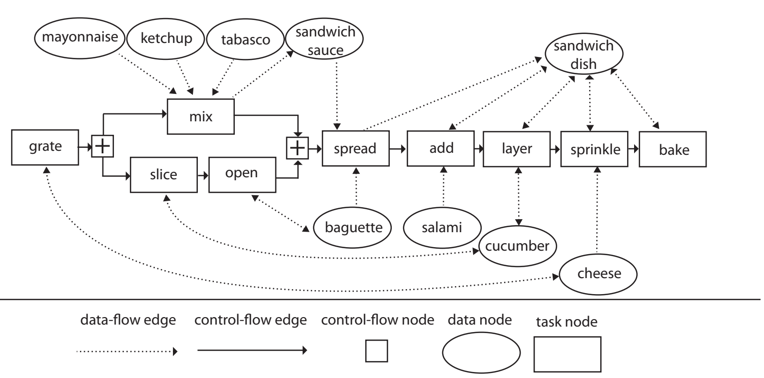


Fig. 2. Example workflow from the domain of cooking.

preparation step (saute) and the ingredients consumed (onion, green pepper) are identified and transformed into a workflow fragment. A stepwise extraction of the entire process description thereby constructs a complete workflow.

The developed extraction methods are able to identify the activities of the process [27], organizing them in a control flow [24], and enriching the control flow by data flow information [26]. For the latter, we additionally investigated an alternative approach to complete missing data-flow information [21] by learning completion operators from a set of revised workflows within the repository. The framework implements a pipe-and-filters architecture. Different extraction steps can be implemented as independent components (filters), which can be composed to an extraction sequence (pipe). Consequently, this allows the flexible reuse and exchange of filters. For the basic linguistic analysis of the textual descriptions, methods from natural language processing have been applied. We used the developed framework to extract a repository of cooking workflows from 35,000 online recipes. The source code of the workflow extraction framework as well as the repository are available for download under open source licenses[1].

5 Similarity-based Workflow Retrieval

For reusing the extracted procedural experiences, the workflow repository is searched for the best matching workflow using similarity-based retrieval methods. In order to capture the scenario or problem situation, a specific workflow query language POQL [20] was developed. The query may include single workflow elements as well as entire workflow fragments (e.g., sub-workflows), which are either marked as desired or undesired. Furthermore, also generalized workflow elements such as generalized tasks and generalized data items can be specified.

POQL can then be used to trigger a similarity-based retrieval for the workflow best matching the requirements and restrictions defined, for which several methods have been developed. Most basically, we developed a semantic similarity measure for semantic workflows [2] which is based on a workflow ontology. The semantic similarity of workflows is defined as an optimization problem for the mapping of workflow elements from the query to the mostly similar elements of case workflow. Various search algorithms and respective heuristics have been developed to efficiently compute this similarity [2]. As an alternative approach to the developed semantic similarity measures, we investigated similarity measures based on the trace index of a workflow [25]. A trace index is created by analyzing all potential execution traces. Similarity of workflows is then computed by comparing the trace indices of workflows.

Moreover, several methods have been developed aiming at improving the efficiency of similarity search within the repository, which is particularly important when the workflow repository grows. For this purpose, a two-level retrieval method has been developed [6]. Additionally, we investigated new methods for workflow clustering based on the developed semantic similarity measures [4]. In particular, we developed various algorithms that explore this cluster structure as an index structure for retrieval [14].

---

[1] `www.wi.informatik.uni-frankfurt.de/index.php?option=com_content&view=article&id=126`

6 Automatic Workflow Adaptation

We aim at supporting the users in situations in which the best matching workflow from the case base does not sufficiently fulfill the query. This requires that the workflow is automatically adapted according to the given restrictions and requirements, i.e., workflow elements or fragments are added or deleted according to the particular needs.

For that purpose, we developed several workflow adaptation methods. Since such adaptation methods usually require a significant amount of domain-specific adaptation knowledge, we additionally developed new methods that allow to automatically learn the required adaptation knowledge from the workflow repository. Hence, we distinguish between a learning phase of adaptation knowledge and a problem solving phase in which for a given query the best matching workflow is adapted such that it matches the particular problem scenario at best (see Fig. 3). The developed adaptation methods can mostly be classified into *transformational adaptation*, *compositional adaptation* and *adaptation by generalization* [9].

More precisely, we developed two *transformational adaptation* methods, which differ in the representation of the adaptation knowledge. In both approaches, adaptation of workflow cases is performed by chaining several transformation steps $w \overset{\alpha_1}{\to} w_1 \overset{\alpha_2}{\to} \ldots \overset{\alpha_n}{\to} w_n = w'$ which iteratively transform the retrieved workflow w towards the adapted workflow w'. This process is a search process with the goal to achieve an adapted workflow which is as similar as possible to the query. Thus adaptation is considered an optimization problem. In *case-based adaptation* [10] the individual transformation steps are represented as so called *adaptation cases* which are learned automatically from the workflow repository [12]. An adaptation case represents a particular previous adaptation scenario by capturing the information about how to transform a particular

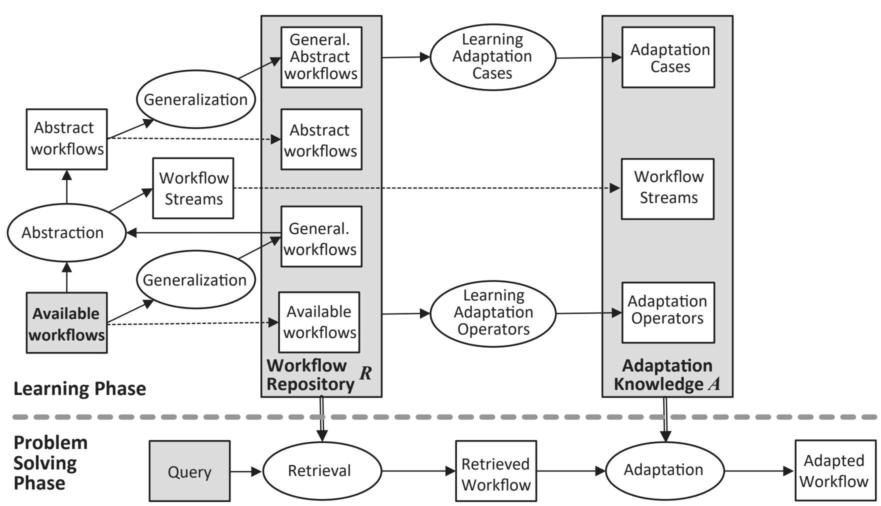


Fig. 3. Integration of adaptation approaches.

origin workflow to a corresponding goal workflow. It can be applied if it matches at a certain position within the workflow to be adapted. The *operator-based adaptation* [19] represents the individual transformation steps as so called *workflow adaptation operators*. They are denoted in a STRIPS-like manner, i.e., by specifying a fraction of the workflow to be deleted and a fraction to be added to the workflow. A learning algorithm was also developed that allows to automatically acquire adaptation operators from pairs of similar cases from the workflow repository.

In addition, we developed a method for *compositional and hierarchical adaptation*. It is based on the idea that each workflow can be decomposed into meaningful sub-workflows called *workflow streams* [16]. Such workflow streams can be automatically discovered from the workflow repository. Workflow streams represent valuable adaptation knowledge which is used as "chunks" that can be inserted or used as replacement during compositional adaptation. Compositional adaptation is also implemented as a search process, but it replaces larger portions of a workflow than the transformational adaptation approaches. In addition, workflow streams provide a means for abstraction. An abstracted workflow, is a structurally simplified workflow, i.e., containing fewer nodes or edges. *Abstraction* is achieved by replacing each discovered workflow stream in a case by a single abstract task. As further background knowledge for abstraction, domain-specific abstraction rules have been introduced, describing how to map a sub-workflow to a domain-specific abstract task linked with an appropriate semantic description from the domain ontology. The abstraction rules consist of elementary abstractions such as sequential abstraction, block abstraction, and elimination. Abstraction can be performed hierarchically, i.e., a rule can abstract also non-primitive tasks. During problem solving, abstract cases (which are also stored in the workflow repository) can be retrieved and reused by refining the occurring abstract tasks, e.g. by using workflow streams as refinement operators, best suited to the current query.

Finally, *generalization and specialization* was investigated as a third adaptation approach [18]. A *generalized workflow* is structurally identical to the base workflow but the semantic descriptions of task and data items are generalized. We generalize a workflow by considering a set of similar workflows as training samples and employ the ontology as generalization hierarchy from which generalized semantic descriptions are selected. The computed generalized cases are added to the workflow repository. During problem solving, adaptation is performed by specializing a previously generalized workflow in a manner, best suited to the current query.

The adaptation methods just described have also been integrated [17] as shown in Fig. 3. In particular adaptation cases and adaptation operators can be learned not only from the available concrete-level cases, but also from cases resulting from abstraction or generalization. Also, case generalization can be performed on top of abstraction. As a consequence, a large spectrum of possible ways arise for learning adaptation knowledge. As a result of the integrated learning process, the workflow repository R consists of four type of cases: 1. the available concrete cases, 2. generalized cases, 3. abstracted cases, and 4. generalized abstract cases. The adaptation knowledge A consists of adaptation operators, adaptation cases, and streams. During problem solving, i.e., when a new workflow for a given new query must be determined, the most similar (generalized/abstract) workflow from the workflow repository R is retrieved. Then, during adaptation the available adaptation knowledge from A is applied in a local search process in order to achieve an adapted workflow which is most similar to the query.

The availability of the previously introduced adaptation methods changes the utility of the workflows stored within the repository. A workflow with a lower similarity value during retrieval might more likely be adaptable to the particular problem situation. Hence, we developed a novel approach for the adaptation-guided retrieval of workflows [5], aiming at identifying the workflow which can at best be adapted to the particular situation during retrieval. The approach basically assesses the adaptability of the workflows by performing several example adaptations.

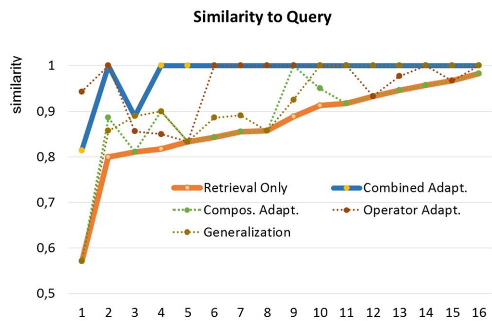
7 Implementation and Experimental Evaluation

The approaches developed throughout the whole project have been continuously integrated in a prototype system called CookingCAKE[2] for participation in the Computer Cooking Contest in 2011 [29], 2012 [3], 2014 [15,28] and 2015 [17]. Using the previously sketched retrieval and adaptation methods, CookingCAKE demonstrates the generation of sandwich recipes considering ingredients and preparation steps that are desired or undesired. A large number of experimental evaluations have been performed which are reported in the papers describing the individual methods. In the following, we show some preliminary experimental results of a preparatory study we performed in the process of the preparation of a more comprehensive systematic trial. In this experiment we used a case base of 60 extracted pasta recipe workflows that have been further improved manually. In a study with human users of CookingCAKE we elaborated 16 realistic queries representing the user's desires for cooking. CookingCAKE was used in various conditions (pure retrieval, use of all adaptation methods in isolation, and the combined adaptation approach) to produce the desired recipe workflow. We compared the system (see Fig. 4) in the various conditions a) by assessing the similarity of the resulting solution workflow to the query and b) by asking the users to assess query fulfillment and quality of the resulting recipes on a 5-point Likert scale. The indicated values for workflow quality and query fulfillment are the difference resulting from adaptation, compared to pure retrieval, thus indicating the impact of adaptation. These initial results indicate that the adaptation methods improve the workflow w.r.t. the degree to which the requirements in the query are fulfilled. On the other hand, workflow quality is decreased to a certain degree. Overall, the combined approach performs best and in particular only leads to a minimal reduction of the quality. These results look promising, but a final assessment and a clear view of the various benefits and shortcomings of the methods can only become substantiated after the final trial is completed.

8 Future Work

The EVER project is currently in its second funding phase (2017 – 2020). During this phase, we aim at working on four novel issues.

[2] https://www.uni-trier.de/index.php?id=40545



Overall Impact on Workflow Quality and Query Fulfillment

Adaptation Approach	Workflow Quality	Query Fulfillment
Combined	-0.17	+1.14
Stream-based	-0.34	+0.25
Operator-based	-0.61	+0.92
Generalization	-0.11	+1.02

Fig. 4. Experimental results: a) similarity comparison, b) user assessment.

– *Adaptation Quality:* While in our previous research, we developed methods that enable the automatic adaptation of workflows by using adaptation knowledge automatically acquired by machine learning methods from workflow repositories, the quality of the adapted workflows is difficult to control. Therefore, we aim at investigating new methods for assessing the quality of automatically adapted workflows as well as methods to assess the impact of each piece of learned adaptation knowledge on the resulting workflow quality. This allows to better control which adaptation knowledge to retain and which to discard.
– *Interactivity:* The retrieval and adaptation methods developed so far are fully automatic, i.e., they adapted a retrieved workflow according to a specified change request (or goal) without further user interaction. However, specifying a workflow goal or even a change request for an existing workflow in sufficient detail turned out to be quite difficult. Therefore, we aim at developing new methods for conversational POCBR [30] that enable fully interactive problem solving involving retrieval and adaptation of workflows.
– *Transfer Learning:* The adaptation methods investigated so far require existing procedural knowledge of significant volume in order to learn enough adaptation knowledge. This makes it difficult to address small or newly emerging domains in which procedural knowledge is still sparse. Therefore, we aim at investigating whether transfer learning methods can be used to improve learning of adaptation knowledge by transferring knowledge from a different, but related domain with substantial procedural knowledge [11].
– *Exploring New Application Domains:* So far, we demonstrated our methods primarily in the domain of cooking workflows. In the second funding period, we aim at broadening the experimental basis for the whole project by exploring workflow and business process model repositories available in existing repository collections. Furthermore, we will expore the field of scientific text mining workflows in more detail.

Acknowledgements. This work was funded by the German Research Foundation (DFG), project numbers BE 1373/3-1, BE 1373/3-3, MI 1455/1-1, MI 1455/2-3.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications 7(1), 39–59 (1994)
2. Bergmann, R., Gil, Y.: Similarity Assessment and Efficient Retrieval of Semantic Workfows. Information Systems 40, 115–127 (2014)
3. Bergmann, R., Minor, M., Schumacher, P., Islam, S.: Cooking CAKE. Presentation and Demonstration at the Computer Cooking Contest at ICCBR 2012 (2012)
4. Bergmann, R., Müller, G., Wittkowsky, D.: Workflow clustering using semantic similarity measures. In: Timm, I. (ed.) KI 2013: Advances in Artificial Intelligence - 36th Annual German Conference on AI. Lecture Notes in Computer Science, vol. 8077, pp. 13–24. Springer (2013)
5. Bergmann, R., Müller, G., Zeyen, C., Manderscheid, J.: Retrieving adaptable cases in process-oriented case-based reasoning. In: Markov, Z., Russell, I. (eds.) Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016, Key Largo, Florida, May 16-18, 2016. pp. 419–424. AAAI Press (2016)
6. Bergmann, R., Stromer, A.: MAC/FAC retrieval of semantic workflows. In: Boonthum-Denecke, C., Youngblood, G.M. (eds.) Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2013, St. Pete Beach, Florida. May 22-24, 2013. AAAI Press (2013)
7. Dufour-Lussier, V., Ber, F.L., Lieber, J., Nauer, E.: Automatic case acquisition from texts for process-oriented case-based reasoning. Inf. Syst. 40, 153–167 (2014)
8. Kendall-Morwick, J., Leake, D.B.: Facilitating representation and retrieval of structured cases: Principles and toolkit. Inf. Syst. 40, 106–114 (2014)
9. Lopez Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., Cox, M.T., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision and retention in case-based reasoning. The Knowledge Engineering Review 20(03), 215–240 (2005)
10. Minor, M., Bergmann, R., Görg, S.: Case-based adaptation of workflows. Information Systems 40, 142–152 (2014)
11. Minor, M., Bergmann, R., Müller, J.M., Spt, A.: On the Transferability of Process-oriented Cases. In: Goel, A.K., Diaz-Agudo, B., Roth-Berghofer, T. (eds.) Case-Based Reasoning Research and Development - 24nd International Conference, ICCBR 2016, Atlanta, USA. Proceedings. Lecture Notes in Artificial Intelligence, vol. 9969, pp. 281–294. Springer (2016)
12. Minor, M., Görg, S.: Acquiring Adaptation Cases for Scientific Workflows. In: Case-Based Reasoning. Research and Development, 19th International Conference on Case-Based Reasoning, ICCBR 2011. LNCS, vol. 6880, pp. 166–180. Springer (2011)
13. Minor, M., Montani, S., Recio-García, J.A.: Process-oriented case-based reasoning. Inf. Syst. 40, 103–105 (2014)
14. Müller, G., Bergmann, R.: A cluster-based approach to improve similarity-based retrieval for Process-Oriented Case-Based Reasoning. In: 20th European Conference on Artificial Intelligence (ECAI 2014), IOS Press. pp. 639–644 (2014)
15. Müller, G., Bergmann, R.: Compositional Adaptation of Cooking Recipes using Workflow Streams. In: Computer Cooking Contest, Workshop Proceedings ICCBR 2014. Springer (2014)
16. Müller, G., Bergmann, R.: Workflow Streams: A Means for Compositional Adaptation in Process-Oriented Case-Based Reasoning. In: Case-Based Reasoning. Research and Development, 22th International Conference on Case-Based Reasoning, ICCBR 2014. pp. 315–329. LNCS 8766, Springer (2014)

17. Müller, G., Bergmann, R.: CookingCAKE: A Framework for the adaptation of cooking recipes represented as workflows. In: Computer Cooking Contest, Workshop Proceedings ICCBR 2015. Springer (2015)
18. Müller, G., Bergmann, R.: Generalization of Workflows in Process-Oriented Case-Based Reasoning. In: Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2015. pp. 391–396. AAAI Press, Hollywood (Florida), USA (2015)
19. Müller, G., Bergmann, R.: Learning and Applying Adaptation Operators in Process-Oriented Case-Based Reasoning. In: Case-Based Reasoning. Research and Development, 23th International Conference on Case-Based Reasoning, ICCBR 2015. pp. 259–274. LNCS 9343, Springer (2015)
20. Müller, G., Bergmann, R.: POQL: A new query language for process-oriented case-based reasoning. In: Bergmann, R., Görg, S., Müller, G. (eds.) Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB, Trier, Germany, October 7-9, 2015. CEUR Workshop Proceedings, vol. 1458, pp. 247–255. CEUR-WS.org (2015)
21. Müller, G., Bergmann, R.: Case completion of workflows for process-oriented case-based reasoning. In: Goel, A.K., Díaz-Agudo, M.B., Roth-Berghofer, T. (eds.) Case-Based Reasoning Research and Development - 24th International Conference, ICCBR 2016, Atlanta, GA, USA, October 31 - November 2, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9969, pp. 295–310. Springer (2016)
22. Plaza, E.: Semantics and experience in the future web. In: Althoff, K.D., Bergmann, R., Minor, M., Hanft, A. (eds.) Advances in case-based reasoning, 9th European conference, ECCBR 2008. Lecture Notes in Artificial Intelligence, vol. 5239, pp. 44–58. Springer, Berlin (2008)
23. Schumacher, P.: Workflow extraction from textual process descriptions. Ph.D. thesis, Goethe University Frankfurt am Main (2015)
24. Schumacher, P., Minor, M.: Extracting control-flow from text. In: Proc. of the 2014 IEEE 15th International Conference on Information Reuse and Integration. pp. 203 – 210. IEEE, San Francisco, California, USA (2014)
25. Schumacher, P., Minor, M.: Towards a Trace Index Based Workflow Similarity Function. In: Lutz, C., Thielscher, M. (eds.) KI 2014: Advances in Artificial Intelligence - 37th Annual German Conference on AI. LNCS, vol. 8736, pp. 225–230. Springer (2014)
26. Schumacher, P., Minor, M., Schulte-Zurhausen, E.: Extracting and enriching workflows from text. In: Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on. pp. 285–292. IEEE (2013)
27. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of Procedural Knowledge from the Web. In: Proc. of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16–20, 2012 (Companion Volume). pp. 739–747. ACM, New York (2012)
28. Spät, A., Keppler, M., Schmidt, M., Kohlhase, M., Lauritzen, N., Schumacher, P.: Goethe-Shaker - Developing a rating score for automatedevaluation of cocktail recipes. In: Proceedings of the Computer Cooking Contest Workshop at the International Conference on Case-Based Reasoning (ICCBR). pp. 101–108 (2014)
29. Walter, K., Minor, M., Bergmann, R.: Workflow Extraction from Cooking Recipes. In: Diaz-Agudo, B., Cordier, A. (eds.) Proceedings of the Computer Cooking Contest Workshop at the International Conference on Case-Based Reasoning (ICCBR 2011). pp. 207–216 (2011)
30. Zeyen, C., Müller, G., Bergmann, R.: Conversational Process-Oriented Case-Based Reasoning. In: Case-Based Reasoning Research and Development - 25th International Conference, ICCBR 2017, Trondheim, Norway. Proceedings. Lecture Notes in Artificial Intelligence, Springer, accepted for publication (2017)

Behavioural Analytics using Process Mining in On-line Advertising

Maria Diapouli[1], Stelios Kapetanakis[1], Miltos Petridis[2] and Roger Evans[1]

[1]School of Computing, Engineering and Mathematical Sciences, University of Brighton, Watts Building, Lewes Road, Brighton, BN2 4GJ, UK
[2]Department of Computing, University of Middlesex
[1]{M.Diapouli, S.Kapetanakis, R.P.Evans }@brighton.ac.uk, [2]M.Petridis@mdx.ac.ukuk

Abstract. Online behavioural targeting is one of the most popular business strategies on the display advertising today. It is based primarily on analysing web user behavioural data with the usage of machine learning techniques with the aim to optimise web advertising. Being able to identify "unknown" and "first time seen" customers is of high importance in online advertising since a successful guess could identify "possible prospects" who would be more likely to purchase an advertisement's product. By identifying prospective customers, online advertisers may be able to optimise campaign performance, maximise their revenue as well as deliver advertisements tailored to a variety of user interests. This work presents a hybrid approach benchmarking machine-learning algorithms and attribute pre-processing techniques in the context of behavioural targeting in process oriented environments. The performance of our suggested methodology is evaluated using the key performance metric in online advertising which is the predicted conversion rate. Our experimental results indicate that the presented process mining framework can significantly identify prospect customers in most cases. Our results seem promising, indicating that there is a need for further workflow research in online display advertising.

Keywords: Process mining, Process Oriented Workflows, Classification, Online Display Advertising.

1 Introduction

According to statistics published by the Internet Advertising Bureau online advertisers in the UK have spent more than 8.6 billion UK-pounds in 2016 on behavioural targeted advertising a figure which grew 16.4% compared to 2015. The estimate represents steady growth rates of about 20% from 2010 through 2016 [3]. Behavioural targeting and customer prospecting are both promising and challenging aspects in display advertising. Promising since the more information of user behavioural activity exists the better targeted advertisements could be delivered to end users and challenging since display advertising is a rather complex ecosystem which involves multiple interested parties such as end users, advertisers, publishers, and ad platforms. The size of data generated and collected from any involved parties is significantly large: Billions of websites requests every day trigger millions of advertisements that are finally displayed to millions of users.

Digital advertisers attract increasing traffic on their websites aiming for certain user marketing actions, more commonly, accomplishing an online purchase. This action is recorded as a conversion. There are two ways for viewing an advert upon arrival on an affiliate ad-friendly website. Firstly, by clicking on the advert and immediately buying and/or by viewing an advert and waiting for a future return and a possible purchase. The journey of a user throughout several websites can be represented as a series of events with intermediate temporal durations. This can be interpreted into a "workflow" of variant length which may or may not convert at its final stages. Petridis et al. [19] have shown that workflow behaviours with such a distinct event-duration coupling can be formalised over a general theory of time [20], be graph-represented, monitored [21] and explained [22] effectively using Case-based Reasoning techniques [23].

Our research questions on top of the online marketing business model are twofold – One: which metric features in terms of evaluating an online campaign performance are mostly important and -Two: based on the set of identified metrics what is the profile of an ad viewer who is keen to make a purchase. In such way by analysing and classifying past behavioural observations among ad viewers, could allow marketers to identify future prospect customers more effectively.

The work we present in this paper handles a challenging area in the online display advertising marketplace, this of customer prospecting. Customer prospecting identifies web users who are likely to purchase a product after seeing an advertisement. We developed a process mining methodology based on an advertising campaign implemented by an ad network provider. We collected and analysed campaign data that contained audience demographic information and audience behavioural segments to predict whether a user who had no previous seen an advert is likely to convert. The goal of this research was to increase an individual advertising campaign performance by augmenting its CPA ratio.

This paper is structured as follows: Section 2 presents the context of search engine advertising and its online display landscape, section 3 will describe our adopted process mining methodology. Additionally, the imbalanced problem of conversion rate will be explained and our approach to the class imbalance problem will be analysed. Section 4 will present a series of empirical experiments for selecting the best performing classification algorithm. Finally, a discussion upon our experiment results will be presented in section 5.

2 Related Work

2.1 Search Engine Advertising

The application of statistical algorithms and process mining methodologies is widely applied in search engine advertising. Its outcomes could be observed from user-relevant textual advertisements placed next to search results as they come from several search engines. Choosing the most relevant ad for a user query and the optimal place in which it is displayed could affect significantly the probability for a user to click on that chosen ad.

For any adverts with already known Click-Through Rate (CTR) historical information, CTR could be estimated empirically by dividing the number of impressions over clicks. In any other case when a new ad, with no historical information, is going to be displayed to a random user; a major challenge emerges: This is mainly in terms of identifying a suitable advert where a user would be “tempted to click”. Richardson et al. [9] answer this challenge by predicting the CTR for a new ad with no prior historical information. Based on ad text information only (title, body, search keywords, display URL, impressions, clicks, landing URL) logistic regression can produce an accurate model ad CTR prediction. Research in the area has shown that decision rules [7] can be produced for predicting the CTR for unseen ads, from data that contain information regarding advertisements, query terms and URLs. Clustering techniques could also be used to improve the keywords CTR for rare or new keywords [8] based on generation of clusters of related keywords. This can be applied by if different search keywords have a different likelihood of receiving a user click [8].

2.2 Online display advertising landscape

Online Display Advertising is a highly congested and convoluted environment involving an extended range of vendors, services and high volumes of transactions. Its landscape mainly comprises workflows of advertisers, ad agencies, web-users and publishers. Advertisers set up product or service workflows in publisher web sites, also known as inventories, with the aim to attract as many web-users as possible. This is achieved by providing rich and engaging ad-context to the most receptive online audience. In return the end users will click on the ad and will be redirected on the advertiser website to purchase potential product(s).

The complexity of achieving such desired actions has led to the development of new display parties, these of Ad-exchanges and Ad-networks. Both could perform better on accessing and controlling inventories. Ad-exchanges are online auction marketplaces (like e-bay) trading in advertisements as their “products”. Ad-exchanges could provide three main services: adverts allocation, prices determination, traffic control [1]. Ad networks provide a variety of excluded services to advertisers such as ad serving, privacy verification, targeting the most suitable audience and advertising campaign reporting. Ad networks could access publishers directly as well as ad exchanges for identifying inventories.

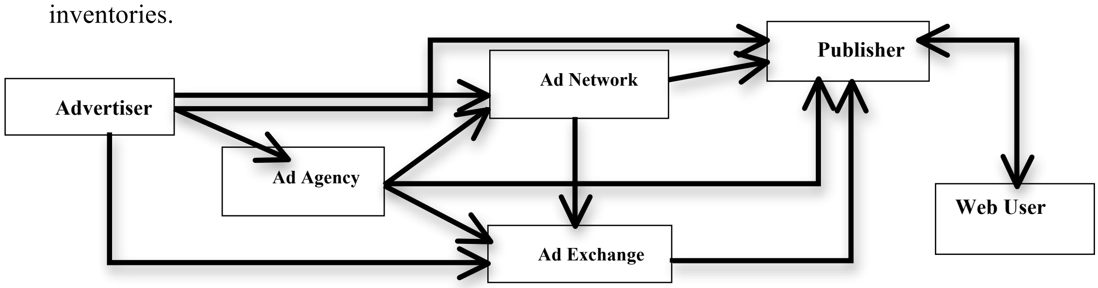


Figure 1: Key stakeholders of an online-advertising workflow

Measuring the revenue and the effectiveness of online display advertising campaigns is achieved through three prevalent pricing models. These are: Cost per impression (CPM), Cost per click (CPC) and Cost per action (CPA). CPM ratio is popularly used for brand recognition campaigns where a fixed cost is charged to advertisers based on the number of displays of advertisements. CPC ratio was introduced to build the advertisers confidence upon their return on investment(s). Advertisers pay publishers when a web user clicks on an advertisement without considering the number of impression displays. However, since most advertisers are retailers the actual advertising benefit derives from the commercial transaction within their websites [2]. CPA metrics is used to serve this purpose. According to the Interactive Advertising Bureau report (2016) in the US market, 65% of online advertising transactions share was attributed on the customer performance models (CPC, CPA), while the second on the list was the CPM model with 33%. Hybrids of impression and performance models reside in 2% of online advertising transactions [3].

According to Lewis and Reiley (2014) [16] the effect of online advertising on sales is not fully associated with CTR. Their collaboration with Yahoo!Research and an eminent retailer had reported that 78% of the lift in retailer sales was originated from users who had viewed ads but had not clicked them, while only 22% was attributed to those who had clicked. In our research work we found that the online advertising campaign had substantial impact on the users who merely viewed the ads. Based on thorough analysis we identified that impressions are more strongly correlated to conversions than clicks. Most interestingly, clicks had a very trivial correlation (correlation = 0.00000115) with conversion. These findings suggest that the most meaningful metric for evaluating campaign performance is conversions instead of clicks.

3 Methodology

Customer prospecting is related with predictive modelling in process mining terms. Predictive modelling, also referred as supervised data mining, aims to predict the probable future event based on previous historical knowledge [10]. The appropriate selection of data samples is important for effective analysis and prediction based on underlying patterns [13]. In this work decision trees and kNN were preferred over the commonly used logistic regression and collaborative filtering classification methods. Decision trees have been shown as effective in building profiles for the web users who have converted in the past and then predict whether a new web-user is likely to convert [17]. Decision trees although powerful in expressing continuous and categorical inputs, they seem to fall out when there is a mixture of continuation and categorical type data. Thus, kNN has seemed more appropriate since it performs better with continuous data [18]. For predictive modelling decision trees seem most powerful and prevalent tools [15] compared to logistic regression and collaborative filtering since:

-Logistic regression could be used to examine the model's exponentials of the coefficients to explore which user attributes affect the likelihood of conversion. In such way, we would be able to explore the necessary coefficients but be unable to explore the underlying ruleset which could indicate and predict online “target” users willing to convert.

-Collaborative Filtering was also considered since it has been proven effective in finding prospect customers based on past customer behaviours (training samples) [14]. In such way, a successful model would be retrained on a regular basis to include recent user activity information. However, in our investigated dataset such information was not available and this research was not able benefit from “live ad-feeds” including: Ad ids, ad-width, ad-height, visibility time viewed, format, etc.

Therefore, based on the limitations of the dataset and the model target audience, our selection methodology was based on data-mined patterns for ruleset generation to understand and predict successful (or not) online user-conversions.

3.1 The Data

The dataset used in this work was retrieved from real advertising campaigns as conducted by a UK-based advertising company. The company was using a No-SQL distributed database management system (DDMS) based on Apache Hadoop. Any marketing data was extracted from DDMS and stored in tab separated files (tsv) textual format for further processing and analysis. During a one-week campaign 20 million impressions were displayed to web users approximately, a figure which was increased exponentially over more campaigns and longer campaign times or series of campaigns.

The used dataset comprised three distinct types of ad-logs, described as: impressions, clicks, and conversions. Any available log data were organised and aggregated based on the user id feature on a row-per-record basis. A sample of the feature description and example for each column of the ad log data are presented in Table 1. Each record contained three information types: (i) Behavioural data (columns 1,2,6,7,8,9) (ii) Interest profiles (columns 1,2,4,6,7,8,9,10,11,12) and (iii) Intent profiles (columns 1,2,3,6,7,8,9).

Column	Description	Example
1	User id	156780d5128b4c1cb1bc5652ebcadd2d
2	Location data	UK, London
3	Interest profiles	shopping
4	Intent profiles	home and garden
5	Previous browsing activity	https://www.gumtree.com/p/dinnerware-crockery/tea-set/1183263978
6	Device	tablet
7	Browser	mobile safari
8	Operating system	android2
9	Language	English
10	Impressions	19
11	Clicks	1
12	Conversions	1

Table 1: Sample of a data ad-log format.

The dataset used in this experiment was gathered as a day-campaign and it consisted of 3,425,119 impressions that were displayed on 3,407,293 users. Among them 8,082 users clicked on the displayed advert (click response rate 0.24%) and 913 converted (convert response rate 0.03%). Due to the very sparse number of response rate both for click and conversion the data were highly skewed. To overcome this limitation of imbalanced data a sampling technique was adopted and will be discussed in the next section. The size of the data set had 3,407,293 observations. Each observation characterised a web user and was described by 46 independent features and 1 dependent feature (sample of log data are shown in Table 1). The independent features were both categorical and nominal. These features were related to a user's browsing history (URLs that user has visited in the past). The dependent feature was a nominal one-named conversion rate which illustrated whether a user had made a purchase in the past.

3.2 Our Approach

The process of Knowledge Discovery in Databases (KDD) was adopted as methodological approach on this case study, where process mining was the gist in the overall process [11]. The experiment went through all the steps of KDD. The data were extracted from the DDMS based on the process mining project specifications to ensure consistency and completeness. In the data transformation phase, categorical values where transformed on numerical ones to adhere to the algorithm specifications. Additionally, since the response rate for conversion was a small number, only 0.03% of the total dataset, a method that modified data values and derived new fields from response rate for conversion was adopted. A new field was created which contained two values for the conversion field: zero (indicated that a user did not convert) and one (indicated a successful user conversion). However, this new field did not overcome the very low percentage of conversion rate. This was regarded a challenge since a high-performance classifier was required to have an accurate model. In such: training data should be evenly distributed between conversion and no-conversion values. In any other case the classifier could be biased since it would try to achieve the overall classification accuracy by identifying mostly the majority class (conversions) and would overlook the minority one (no-conversion). This would offer little contribution to the model accuracy [4]. Our approach in balancing the classifier will be described in section 4.2.

4 Experiments and Evaluation

4.1 Modelling and imbalanced data sets

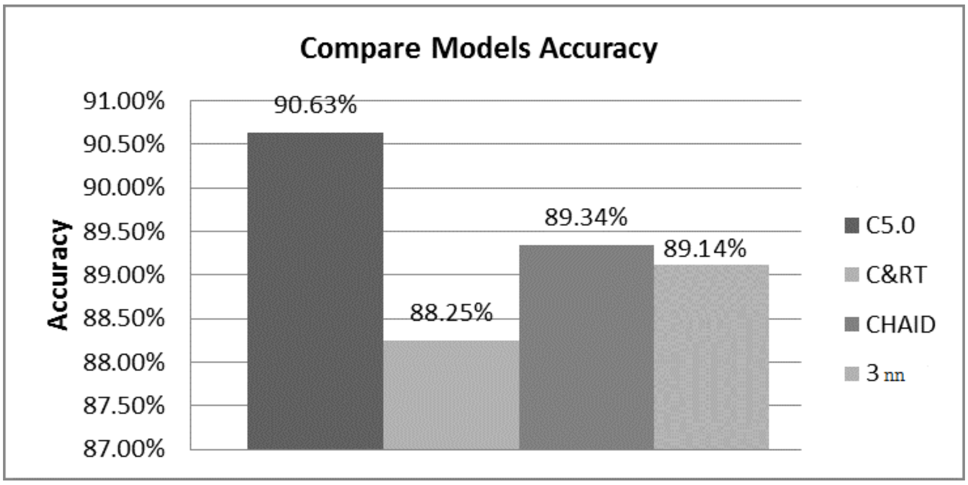
For this work, IBM SPSS Modeler 18.0 was used for all experiments. Different algorithms were assessed to benchmark the most appropriate and accurate dataset for classification and prediction. The data set was separated into training and testing set to build and evaluate our decision tree model with a 70 - 30 split rate respectively. In the training phase, the model was processed by using the training set and then tested to evaluate our model's accuracy

To overcome the problem of heavily imbalanced data two approaches were considered: a) using over-sampling and b) using under-sampling. In the over-sampling approach, the training set was populated with replicated data that belonged to the minority class until the training set was balanced [6]. The information remains the same but the misclassification cost of the minority class is increased.

In the under-sampling approach data from the majority class were removed to balance the training set [5]. For our experiments, the under-sampling approach was used where cases from the dependent feature conversion rate were randomly eliminated.

4.2 Comparing the performance of different algorithms

Different decision tree algorithms have been selected for searching patterns in data as well as kNN with k =3 for more accurate classification of continuous data attributes. This process included deciding which algorithm provided the lowest average classification error. The selected algorithms were: classification and regression tree (C&RT), Chi-squared automatic interaction detector (CHAID), C5.0 and 3NN. The performance results of these algorithms as applied on the test set are shown in Graph 1.



Graph 1: Summary of results

	C&RT	Chaid	C5.0	3NN
Converter True Positives (Hit) Rate	93.36%	96.48%	99.02%	91.23%
Non-Converter True Positives (Hit) Rate	88.24%	89.34%	90.62%	89.32%
Accuracy	88.25%	89.34%	90.63%	89.14%
Sensitivity	93.36%	96.48%	99.02%	90.15%
Specificity	88.24%	89.34%	90.62%	88.14%

Table 2: Accuracy Measures and Alternatives

Table 2 illustrates a higher likelihood (90.63%) for C5.0 to predict the event for someone converting on an advertisement compared to the other baselines (88.25% for C&RT, 89.34% for CHAID and 89.14% for 3NN). In Table 2 C5.0 is showing 99.02% sensitivity (the portion of users that were correctly predicted to convert) and 90.62% specificity (the portion of users that did not convert and were successfully predicted) which accounts for the overall accuracy of 90.63%. The sensitivity and specificity measures are used to ascertain the model validity and accuracy [12].

4.3 Comparing the performance of the different data sets

Typically, the performance of machine learning algorithms is evaluated using predictive accuracy. The evaluation and interpretation of the mined patterns in terms of reliability and accuracy of the derived rules have taken place in the evaluation phase.

We performed our experiments using 10-fold cross validation. The original dataset was randomly divided into ten (10) subsets. Each time, one of the 10 subsets was used as the test set and the other 9 subsets were combined to form the training set. For the conversion field, there were two classes, the positive class, assigned as 1, that comprised "converted" users and the negative class consisted of "no converters", assigned as 0.

Partition	Training Set	Percentage Training Set	Testing Set	Percentage Testing Set
Correct	1,482	93.09%	3,088,032	90.67%
Wrong	110	6.91%	317,669	9.33%
Total	1,592		3,405,701	

Table 3: Model Accuracy

Table 3 illustrates the correct and wrong prediction of our model.. The rows defined by actual values and columns defined by predicted values, with the number of records having that pattern in each cell.

4.4 Evaluating the model performance with bootstrap aggregation

The data for building decision trees with C5.0 algorithm models were re-sampled using a bootstrap aggregation technique to form several pairs of training and testing data sets. A decision tree model was developed for each pair of training and testing data sets. Predictions from any individual decision trees were merged via a voting system which led to the highest possible accuracy for the final model (ensemble) predictions. It was observed that similar models were generated throughout ensemble learning. This was evidence that the chosen algorithm was stable throughout the dataset.

5 Conclusions

In this paper, we demonstrated process mining as an effective tool for direct marketing which can improve online marketing campaigns. Most the existing research in this area so far focuses on computational and theoretical aspects of direct marketing though little efforts have been put on technological aspects of applying process mining in the process of direct marketing. The complexity of process mining models makes it difficult for marketers to use it, hence we outlined a simplified framework to guide marketers and managers in making use of process mining methods and focus their advertising and promotion on those categories of people to reduce time and costs. We explained the steps and tasks that are carried out at each stage of the process mining framework and showed some examples of the type of predictive efficiency that can be achieved using the proposed approach. This has shown that substantial gains can be achieved by adopting this pragmatic and exploratory approach to predict user behaviour in on-line advertising.

This work has shown capable of dealing with the uncertainty underlying within behavioural data as on-line advertising experts have noted that user behaviours can vary significantly. Our suggested approach seems capable of dealing with more complex online advertising models and thus our future directions will focus on more complex, variant and fuzzy attributes.

The results obtained so far, are promising and encourage us to continue experimentation with more sophisticated models or other algorithms to further improve the performance of the system. It seems sensible to experiment with the following settings in future work:

– introducing the temporal dimension to our model to apply time series analysis techniques to build the model
– combining the model with content-based approach
– additional category-based and continuous-based attributes specifying the times spent on each of the categories, with possible division into work-days and week-days, for example a different choice of categories.

As future work, we will incorporate more user and publisher information obtained from third party media providers into data hierarchies to improve model prediction.

Acknowledgement

This work was funded by Innovate UK (formerly known as Technology Strategy Board).

References

1. Balseiro, S., Feldman, J., Mirrokni, V., Muthukrishnan. S., Yield optimization of display advertising with ad exchange. In ACM Conf on E-Commerce, (2011)
2. Moona, Y., Kwonb, C.: Online advertisement service pricing and an option contract, Electronic Commerce Research and Applications, Elsevier (2011)
3. Interactive Advertising Board (IAB). Internet Advertising Revenue commissioned report, (Updated June 2016), https://iabuk.net/, accessed May 2017
4. Visa, S., Ralescu. A., Issues in mining imbalanced data sets - a review paper. Midwest AI and Cognitive Science Conference, 67-73 (2005)
5. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection, International Conference on Machine Learning, 179–186 (1997)
6. Ling, C., Li, C.: Data mining for Direct Marketing: Problems and Solutions Proc. Of the Fourth Inter. Conf. on Knowledge Discovery and Data mining, AAAI Press, 73-79, (1998)
7. Kotlowski, W., Debmbsczynski, K., Weiss, D.: Predicting ads click through rate with decision rules. In Workshop on Target and Ranking for Online Advertising, WWW 08, (2008)
8. Regelson, M., Fain D. C.: Predicting click-through rate using keyword clusters. In Electronic Commerce (EC). ACM, 2006.

9. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. International conference on World Wide Web, 521–530, (2007).
10. Apte, C. V., Hong, S. J., Natarajan, R., Pednault, E. P. D., Tipu, F. A., Weiss, S. M.: Data-intensive analytics for predictive modelling, IBM journal of research and development , 47(1) (2003)
11. Chapman, P., Clinton, J., Kerber, R., Khabaza T., Reinartz, T., Shearer, C., Wirth, R., 1999. CRISP-DM 1.0 Step-by-step data mining guide. Crisp DM Consortium, (Updated 2010), https://www.the-modeling-agency.com/crisp-dm.pdf, accessed April 2016
12. Han, J., Kamber, M., Kaufmann, M.: Data mining: Concepts and Techniques (2nd Edition) (2006)
13. Kim, Y.: Toward a Successful CRM: Variable Selection, Sampling, and Ensemble, Decision Support Systems, 41(2), 542-553 (2006).
14. Anastasakos, T., Hillard, D., Kshetramade, S., Raghavan, H.: A collaborative filtering approach to ad recommendation using the query-ad click graph, ACM conference on Information and knowledge management, 1927-1930 (2009).
15. Reddy, C.S., Vasu, V., Kumara Swamy Achari B.: Effective Decision Tree Learning, International Journal of Computer Applications, 82(9), (2013).
16. Lewis, R.A., Reiley, D.H.: Online ads and offline sales: measuring the effect of retail advertising via a controlled experiment on Yahoo!, Quantitative Marketing and Economics, 12(3), 235-266 (2014)
17. Diapouli, M., Petridis, M., Evans, R., Kapetanakis, S.: An Industrial Application of Data Mining Techniques to Enhance the Effectiveness of On-line Advertising. In Proceeding of Thirty-sixth SGAI International Conference on Artificial Intelligence, AI 2016, pp.395- 401 (2016)
18. Shah, Z., Mahmood, A. N., Orgun, M. A., Mashinchi, M. H.: Subset Selection Classifier (SSC): A Training Set Reduction Method. In proceedings of the 16th International Conference in Computational Science and Engineering (CSE), 2013 IEEE, pp. 862-869 (2013)
19. Petridis, M., Ma, J., Knight, B.: Temporal model for business process. Intelligent Decision Technologies. 5 (4), pp. 321 – 332 (2011)
20. Ma, J., Knight, B.: A General Temporal Theory, the Computer Journal, 37(2), 114-123 (1994).
21. Kapetanakis, S., Petridis, M.,Knight, B.,Ma, J.,Bacon,L. : A Case Based Reasoning Approach for the Monitoring of Business Workflows, 18th International Conference on Case-Based Reasoning, ICCBR 2010, Alessandria, Italy, LNAI (2010)
22. Kapetanakis, S., Petridis, M., Knight, B., Ma, J.,Bacon, L.: Providing Explanations for the Intelligent Monitoring of Business Workflows Using Case-Based Reasoning, in workshop proceedings of ExACT-10 at ECAI 2010, Lisbon, Portugal (2010)
23. Kapetanakis, S., Petridis, M.: Evaluating a Case-Based Reasoning Architecture for the Intelligent Monitoring of Business Workflows, in Successful Case-based Reasoning Applications-2, S. Montani and L.C. Jain, Editors. 2014, Springer Berlin Heidelberg. p. 43-54.

Case Representation and Similarity Assessment in a Recommender System to Support Dementia Caregivers in Geriatric and Palliative Care

Sara Nasiri and Madjid Fathi

Department of Electrical Engineering & Computer Science, University of Siegen
Institute of Knowledge Based Systems & Knowledge Management
Hölderlinstr. 3, 57076 Siegen, Germany
sara.nasiri@uni-siegen.de
fathi@informatik.uni-siegen.de

Abstract: In this paper, a case-based reasoner uses International Classification of Functioning, Disability, and Health framework of WHO (ICF codes) and medical expressions to create keyword association profiles. Dementia Case-based Learning Assistant System (DePicT Dementia CLASS) finds significant references and learning materials by utilizing the profile of word association strengths according to the problem description. The purpose of this research is to develop a CBR system for recommending the related references by using the information retrieved from dementia books based on the ICF framework of WHO. Case-based learning assistant system helps users to find their answers in dealing with their problems. It also used their feedback to update cases and for improvement of references. This research proposes a combination of references with the highest value keyword association strengths and collaborative recommendation based on the ranked references by the user's feedback.

Keywords: case-based reasoning, dementia, ICF, palliative care, vocational educational training, adaptation

1. Introduction

A CBR methodology is an approach for the recommendation process in medical applications, and especially in medical assistant systems [1, 2, 3, 4, 5]. CBR applied in various problem-solving domains, and it is appropriate in medicine to integrate the system and for explicit experience, cognitive adequateness, a duality of objective/subjective knowledge, and to extract subjective knowledge [6].
"Dementia encompasses a range of neurological disorders characterized by memory loss and cognitive impairment. In 2015, almost 47 million people worldwide were estimated to be affected by dementia, and the numbers are expected to reach 75 million by 2030, and 131 million by 2050, with the greatest increase expected in low-income and middle-income countries [7]." Therefore, the World Health Organization (WHO) in 2012 and 2015, presented reports that Alzheimer's Disease (AD) and other dementias should be regarded as a global public health priority [7]. International Classification of Functioning, Disability, and Health (ICF) [8] is utilized and developed in different projects for disabilities and health problems. "This was believed necessary due to the

complexity of using a large number of ICF codes needed clinically to classify a person's functioning. Indeed, the maximum number of codes per person can be 34 at the one digit level (eight body functions, eight body structures, nine performance and nine capacity codes). At the second level the number of codes is 362; and, at more detailed levels, these codes total up to 1,424 items [8], [9]." ICF framework is also applied in dementia for matching older adults with dementia and technology. To illustrate the use of the ICF in the clinical management of individuals with dementia [10]. To analyze the communication disorders in Alzheimer [11], and to analyze the prevalence of functional impairments, activity limitations and participation restrictions [12].

Textual case-based reasoning (TCBR) is "a subfield of CBR concerned with research and implementation on case-based reasoners where some or all of the knowledge sources are available in the textual format [13]". It aims to use the textual knowledge with an automated/semi-automated approach for problem-solving. Over the years, there has been significant progress addressing the way of bringing textual knowledge sources into the structured case bases [14], [15], [16]. Bousbahi and Chor proposed a system which is called MOOCs-Rec that recommends appropriate courses of Massive Open Online Courses (MOOCs) from different providers in response to a specific request of the learner [17].

The main objective of DePicT Dementia CLASS is to develop DePicT CLASS concept by enrichment of cases with dementia learning materials (e.g. reference images and textbooks). DePicT CLASS is a case-based learning assistant system to detect and predict disease using image classification and text information [18]. DePicT Dementia CLASS is used and updated by caregivers and domain experts. It enables caregivers and patients' relatives to find their learning materials and references which address the problems that they are looking for. Therefore, searching and finding the appropriate learning materials is significantly requested. Although the increasing prevalence of dementia poses a major challenge for global health at multiple levels [19], CBR is applied in the care of AD patients from 2001 [20].

In this paper, we have another objective to help caregivers and patients' relatives by facilitating the finding of dementia references and learning materials with using a DePicT CLASS's retrieval mechanism based on the word association profile of the request. This research has addressed these questions in the paper; how to create structured case representations from texts and how to evaluate the similarity between textual cases.

This paper is structured as follows. Section 1 presents the introduction and related works. Section 2 explains the DePicT CLASS as a preliminary concept. Section 3 presents the case representation of our system based on the ICF parameters. Section 4, first explains the DePicT Profile Matrix and then makes specification on our case retrieval. Finally, section 5 provides conclusions and future work.

2. DePicT CLASS: Preliminary Concept

This section focused on DePicT CLASS and how graphical and textual information is used as the feature to find appropriate references and learning materials within the CBR case matching, selection and adaptation procedure. DePicT CLASS [18] is a complete cyclic CBR system and integrated process of solving a problem, revising the similar solutions and learning from retained experiences which illustrated in Fig. 1. DePicT

Profile Matrix (1) enriched the knowledge base of DePicT CLASS and in the following, DePicT CLASS procedure system explained that how these features are used as attributes. Afterward, how the requested information of the user request is utilized in the case matching, retrieval, and selection process. Finally, the solutions are adapted based on similar cases for the recommendation which are ranked based on the values of keywords.

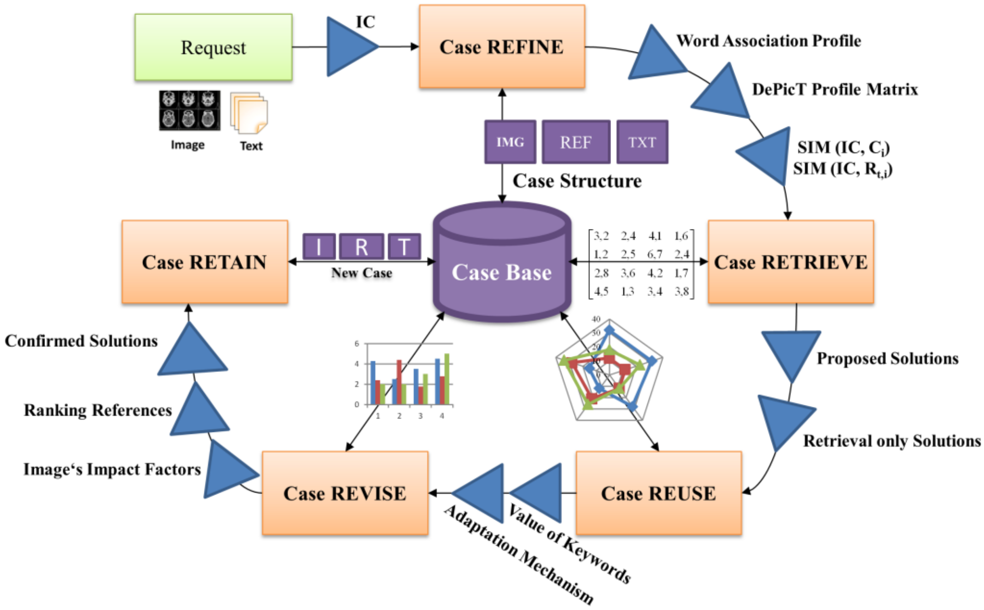


Figure 1: DePicT CLASS Concept [18]

In DePicT CLASS, references and learning materials which are related to the problem and solution are attached to the case as a case description and a case recommendation. Reference images have the word association profile based on the impact factors which are defined by domain experts and tagged to the images. Moreover, DePicT CLASS utilizes collaborative recommendation with tagged keywords, references and learning materials which are ranked by users. The following sections described these principal components of DePicT Dementia CLASS.

3. Case representation based on ICF

Case formation identifies the requested keywords and assigning their values based on DePicT Profile Matrix. This section explained the characterizes a case base and ascertains how incoming cases are refined for retrieving.

Since ICF is inherently a health and health-related classification, it is used as a clinical tool in needs assessment, matched treatments with specific conditions, vocational assessment, rehabilitation and outcome evaluation. It also used as an educational tool in curriculum design and to raise awareness and undertake social action [9]. The structure of ICF is illustrated in Fig. 2 [8].

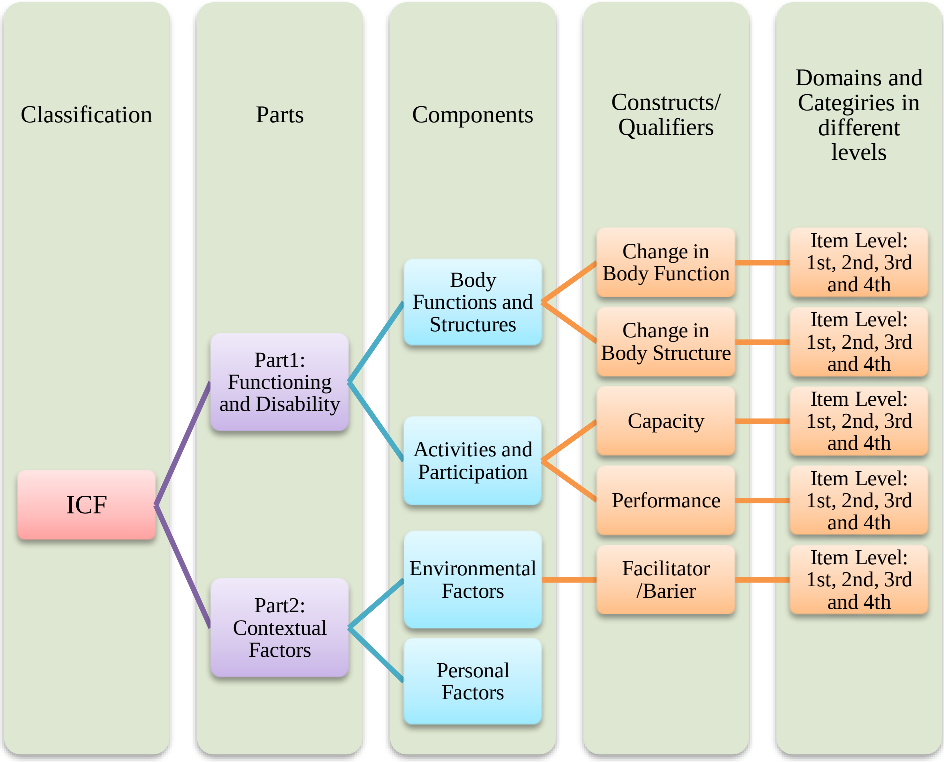


Figure 2: ICF Structure

Scherer et. al. developed ICF codes (111) for dementia with an integrating evidence gathered from preliminary studies that included focus groups of health professionals, a systematic review of the literature, and empirical data collected from patients and caregivers [9]. In this paper, these 111 parameters are utilized as case features. Case representation contains the vectors of ICF word association strengths and ranked common keywords. As shown in Fig 3. case structure includes the identified keywords, problem description, and solution recommendation.

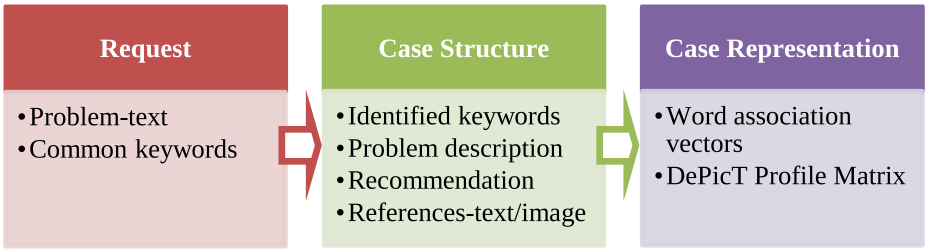


Figure 3: Case Representation

These parameters are searched in dementia, and caregiving books and handbooks e. g. [21, 22, 23, 24, 25, 26, 27] to create the large document as a reference of DePicT Profile Matrix. It is filled based on the ICF words association strength explained in the following section.

4. DePicT Profile Matrix and Case Retrieval

Each case has a word association profile of the main keywords which are defined based on ICF codes and are extracted from case description and case references. The CIMAWA values of the Word Association Strength (WAS) between the case title and case features (identified keywords) are combined in the DePicT Profile Matrix [18].

$$\begin{bmatrix} \mathrm{WAS}(1,1) & \mathrm{WAS}(1,2) & \dots & \mathrm{WAS}(1,n) \\ \vdots & \ddots & & \vdots \\ \mathrm{WAS}(j(t,i),1) & \mathrm{WAS}(j(t,i),2) & \dots & \mathrm{WAS}(j(t,i),n) \end{bmatrix} \qquad (1)$$

WAS(j(t,i),i) is the numeric value of CIMAWA [28] between the title phrase of the case i and jth identified keyword of the t^{th} reference from references and learning materials of case i. The case title phrase is a combination of title's keywords as a text string.

$$WAS(x,y) = CIMAWA^{\zeta}_{ws}(x(y)) = \frac{Cooc_{ws}(x,y)}{(frequency(y))^{\alpha}} + \zeta \frac{Cooc_{ws}(x,y)}{(frequency(x))^{\alpha}} \qquad (2)$$

The composite character of (2) makes it possible to measure symmetric and asymmetric word associations with a damping factor ζ larger than 0. Co-occurrences ($Cooc_{ws}$) of two words x and y in a defined text window size ws are measured in a large document corpus. These damping factors and window size are changed based on the domain. In this research, to have normalized word association strength (between 0 and 1), best results are achieved with utilizing 2 and 0,5 for α and ζ, respectively and text window size is also ten with five words on the right and five words on the left side of selected keyword. This method considers the identified ICF parameters and its synonyms. The word association strength is also calculated based on 37 surveyed dementia books.

To the list of the ICF identified keywords, the word association strength between Alzheimer and memory loss which is the b144 Memory functions from the ICF second level qualifier is calculated based on the description from Alzheimer's Association [29]:

"**Alzheimer**'s is the most common form of dementia**,** a general term for **memory loss** and other cognitive abilities serious enough to interfere with daily life. **Alzheimer**'s disease accounts for 60 to 80 percent of dementia cases. **Alzheimer**'s is a progressive disease, where dementia symptoms gradually worsen over a number of years. People with **memory loss** or other possible signs of **Alzheimer**'s may find it hard to recognize they have a problem."

According to the equation (1) and (2), the frequency, co-occurrence and WAS of these words are calculated as follows:

$$frequency(Alzheimer) = 4 \tag{3}$$

$$frequency(memory\ loss) = 2 \tag{4}$$

$$Cooc_{10}(Alzheimer, memory\ loss) = 1 \tag{5}$$

$$\text{WAS}(Alzheimer, memory\ loss) = \frac{1}{2^2} + 0{,}5\frac{1}{4^2} = 0{,}28125 \tag{6}$$

For an implementation of this formula, first, the library of pdf box [30] is used. The large text which is created based on ICF parameters for each dementia-related diseases defined as a long string, and it is the string array.
In the second step the frequency of keywords and co-occurrence of them in the ten words (five right and five left) window size is calculated. Therefore, the WAS values are calculated for all keywords in each case as cells of DePicT Profile Matrix(was). Each reference has a word association vector with all relevant keywords of the reference. DePicT CLASS checks the similarity of this vector with the new vector (incoming) which is created with the selected input keywords of a user request. We have also DePicT Profile Matrix(w_i) and DePicT Profile Matrix(w_t) for defining the weights in each case and each reference, respectively.

$$\begin{bmatrix} \text{w11} & \cdots \text{w1j} \cdots & w1k \\ \vdots & \ddots & \vdots \\ \text{wi1} & \cdots \text{wij} \cdots & \text{wnk} \end{bmatrix} \tag{7} \qquad \begin{bmatrix} \text{w11} & \cdots \text{w1j} \cdots & w1k \\ \vdots & \ddots & \vdots \\ \text{wtj} & \cdots \text{wtj} \cdots & \text{wqk} \end{bmatrix} \tag{8}$$

Where $w_{ij} = \frac{f_{ij}}{N}$ is the weight of identified keyword j in the case i and f_{ij} is the frequency of word j in the case i and N are the total number of identified keywords including their frequencies in case i. Moreover, where $w_{tj} = \frac{f_{tj}}{Q}$ is the weight of identified keyword j in the reference t and is expressed as follows:
I) f_{tj} is the frequency of word j in reference t and Q is the total number of common keywords between reference t and IC.
II) Moreover, for the reference image t, f_{tj} is the impact factor of word j in the reference t and Q is the sum of impact factors of all common keywords between a reference image and incoming image.
The similarity measurement for comparison of target case or incoming case (IC) and references in DePicT Dementia CLASS is expressed with the following [18]:

$$\text{SIM (IC, } R_{t,i}) = \sum_{i=1}^{n}\sum_{j=1}^{k}\sum_{t=0}^{q}\frac{w_{tj}.w_{ij}\ (R_{t,i}.IC)}{q} \tag{9}$$

where $R_{t,i}$ is the word association profile vector of t^{th} reference from case i.

$$R_{t,i} = (WAS_{1;0;i};\ :::WAS_{j;t;i}:::;WAS_{r;q;i}) \tag{10}$$

Where WASj,t,i is the feature value of the word association strength of word j of t^{th} reference in case i. r is the total number of words in the t^{th} reference of case i, and q is the total number of references in case i.
DePicT Dementia CLASS user interface as shown in Fig 4. consists of a query as a free text, list of selected keywords, result and feedback interfaces. The result part contains

the three most similar cases, adapted references, diagram of DePicT Profile of ICF parameters and ranked references.

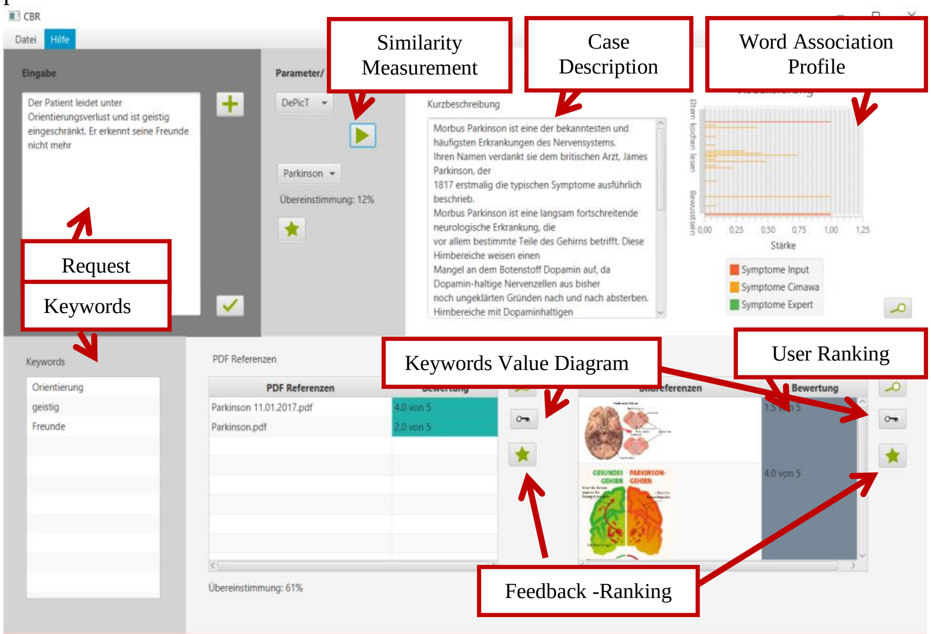


Figure 4: DePicT Dementia CLASS-Case Retrieval:User View

In order to refine the incoming case, IC vector should be created. As an example of the “Requested problem”, user request based on the [29] could be as follows:
“It leads to increasingly severe symptoms, including disorientation, mood and behavior changes; unfounded suspicions about family, more serious memory loss and behavior changes; and difficulty speaking, swallowing and walking.” Each term is as one element in the list of tokens and the example is represented as follows: [It] [leads][to] [increasingly] [severe] [symptoms] [including] [disorientation] [mood] [and] [behavior] [changes] [deepening] [confusion] [about] [events] [time] [and] [place] [unfounded] [suspicions] [about] [family] [friends] [and] [professional] [caregivers] [more] [serious] [memory loss] [and] [behavior] [changes] [and] [difficulty] [speaking] [swallowing] [and] [walking]. Therefore, based on the ICF identified keywords, common keywords from the requested problem are recognized and IC vector is:

$$IC = [0; \text{disorientation}; 0; \text{memory loss}; 0; \ldots; 0; \text{speaking}; \text{swallowing}; 0; \text{walking}] \tag{11}$$
$$= [0; 1; 0; 1; 0; \ldots; 0; 1; 1; 0; 1] \tag{12}$$

After defining the IC, by utilizing similarity measurement (9), the similarity between IC and each case with its references for these five common keywords is calculated. Similarity degrees of all cases are sorted, and the most similar cases are obtained. However, based on the retrieval only approach, each case which has highest similarity degree selected and its solution should be recommended to the user, in DePicT Dementia CLASS, the highest value references and learning materials of the most similar cases (the three highest ones) are selected for the recommendation. Therefore, the DePicT CLASS adaptation mechanism has a combination of value comparison based on the requested word association profiles and manual adaptation based on user collaborative recommendation e.g. learner can rank the best references and learning materials based on their understanding and requirements.

5. Conclusion and Future Work

Developing the DePicT Dementia CLASS is the main contribution of this research. It is a case-based system which uses DePicT Profile Matrix of the association strength between title phrase and identified keywords of cases which are dementia related diseases and ICF parameters, respectively. In this analysis, the dementia references and learning materials with high valued keywords in word association profiles from the most similar cases are recommended. The comparison of word association profiles of selected references including image and text recommends high valued associated references to the problem description of an incoming case. Also, the synonyms of word association profiles are created for each case, based on identified keywords as attributes. During the time of using the system, the learning material is ranked and also updated by caregivers and domain experts.The word association strength of keywords is calculated based on the medical document repositories containing thirty-seven dementia books. In future, the other parameters of caregiving e.g. their challenges and task's difficulties will be considered in the features list. Moreover, for evaluation phase, it will be tested with caregivers instead of test problems from Alzheimer and dementia forums and homepages. This research will be extended to the other aspects of this field to supplement domain expert's knowledge on new, complex and unusual cases.

Acknowledgment We would like to thank our students of the WS 2016/17 project group course for their implementation efforts.

References

[1] B. Winblad et. al., "Defeating Alzheimer’s disease and other dementias: a priority for European science and society," *Lancet Neurology Commission,* vol. 15, pp. 455–532, 2016.

[2] K. Bach and K.-D. Althoff, "Developing Case-Based Reasoning Applications Using myCBR 3," in *Case-Based Reasoning Research and Development,* pp. 17-31, 2013.

[3] M. U. Ahmed, S. Begum, E. Olsson, N. Xiong and P. Funk, "Case-Based Reasoning for Medical and Industrial Decision Support Systems," in *Successful Case-Based Reasoning Application*, S. Montani & L.C. Jain (Eds.), SCI 305, Springer-Verlag Berlin Heidelberg, pp. 7-52, 2010.

[4] R. Schmidt, S. Montani, R. Bellazzi, L. Portinale and L. Gierl, "Case-based reasoning for medical knowledge-based systems," *International Journal of Medical Informatics,* vol. 64, no. 2, pp. 355-367, 2001.

[5] S. Nasiri, J. Zenkert, and M. Fathi, "A Medical Case-based Reasoning Approach using Image Classification and Text Information for Recommendation," in *Advances in*

computational intelligence, Rojas et al. (Eds.): IWANN 2015, Part II, Lecture Notes in Computer Science (LNCS 9095), pp. 43-55, 2015.

[6] N. Choudhury and S. A. Begum, "A Survey on Case-based Reasoning in Medicine," *(IJACSA) International Journal of Advanced Computer Science and Applications,* vol. 7, no. 8, pp. 136-144, 2016.

[7] M.-J. Huang, M.-Y. Chen and S.-C. Lee, "Integrating data mining with the case-based reasoning for chronic diseases prognosis and diagnosis," *Expert Systems with Applications,* vol. 32, no. 3, pp. 856-867, 2007.

[8] WHO, "International Classification of Functioning, Disability, and Health (ICF)," 2001.

[9] M. J. Scherer, S. Federici, L. Tiberio, M. Pigliautile, F. Corradi and F. Meloni, "ICF Core Set for Matching Older Adults with Dementia and Technology," *Ageing Int,* no. Springer Science+Business Media, LLC 2010, pp. 414-440, 2012.

[10] T. Hopper, "ICF and Dementia," *Semin Speech Lang,* vol. 28(4), no. PMID: 17935012 DOI: 10.1055/s-2007-986524, pp. 273-282, 2007.

[11] M. B. Badarunisa, D. Sebastian, R. R. Rangasayee and B. Kala, "ICF-Based Analysis of Communication Disorders in Dementia of Alzheimer's Type," *Dement Geriatr Cogn Disord Extra,* vol. 5, pp. 459-469, 2015.

[12] A. Malara, G. Sgrò, F. Ceravolo, G. Curinga, G. F. Renda, F. Spadea and V. Rispoli, "profiles of icf disability in alzheimer and vascular dementia," *Journal of Aging Research and Clinical Practice,* 2012.

[13] R. O. Weber, K. D. Ashley, and S. Brüninghaus, "Textual Case-Based Reasoning," *The Knowledge Engineering Review,* vol. 30, no. DOI: 10.1017/S0269888906000713, pp. 255-260, 2006.

[14] R. Weber, A. Martins, and R. Barcia, "On legal texts and cases," in *In Lenz, M and Ashley, KD (eds) Textual Case-Based Reasoning: Papers from the AAAI-98 Workshop*, Menlo Park, CA: AAAI Press, pp. 40–50, 1998.

[15] K. Ashley, "Applying Textual Case-based Reasoning and Information Extraction in Lessons Learned Systems," in *AAAI Technical Report*, 2000.

[16] K. Gupta and D. Aha, "Towards acquiring case indexing taxonomies from text," in *In Barr, V and Zdravko, M (eds) Proceedings of the Seventeenth Annual Conference of the International Florida Artificial Intelligence Research Society*, Menlo Park, CA: AAAI, 2004.

[17] F. Bousbahi and H. Chor, "Mooc-rec: A case based recommender system for moocs," *Procedia-Social and Behavioral Sciences,* vol. 195, pp. 1813-1822, 2015.

[18] S. Nasiri, J. Zenckert, and M. Fathi, "Improving CBR Adaptation for Recommendation of Associated References in a Knowledge-based Learning Assistant System," *Neurocomputing*, vol. 250, pp. 5-17, 2017.
DOI: http://dx.doi.org/10.1016/j.neucom.2016.10.078

[19] M. Ienca et. al., "Intelligent Assistive Technology for Alzheimer’s Disease and Other Dementias: A Systematic Review," *Journal of Alzheimer’s Disease,* vol. 56, no. 4, pp. 1301–1340, 2017. DOI: 10.3233/JAD-161037.

[20] S. Marling and P. Whitehouse, "Case-Based Reasoning in the Care of Alzheimer's Disease Patients," *In Lecture Notes in Computer Science,* 2001.

[21] J. W. Ashford and e. al., Handbook of Imaging the Alzheimer Brain, IOS Press BV, 2011.

[22] Y. Mizuno, A. Fisher, and I. Hanin, Mapping the Progress of Alzheimer’s and Parkinson’s Disease, Springer, 2002.

[23] R. Levine, Defying Dementia: Understanding and Preventing Alzheimer's and Related Disorders, 2006.

[24] S. M. Lobo Prabhu, V. A. Molinari, and J. W. Lomax, Supporting the Caregiver in Dementia, Maryland, United States of America: The Johns Hopkins University Press, 2006.

[25] L. Hardman, Dementia (Diseases and Disorders), Gale, Cengage Learning, 2009.

[26] N. L. Mace, The 36-Hour Day: A Family Guide to Caring for People with Alzheimer Disease, Other Dementias, and Memory Loss in Later Life, THE JOHNS HOPKINS UNIVERSITY PRESS, 2006.

[27] C. Turkington, The Encyclopedia of Alzheimer’s Disease, United States of America: ISBN 0-8160-4818-5 (HC), 2003.

[28] P. Uhr, A. Klahold and M. Fathi, "Imitation of the human ability of word association," *International Journal of Soft Computing and Software Engineering(JSCSE),* vol. 3, no. 3, pp. 248-254, 2013.

[29] Alz.org, "Alzheimer's Association, What Is Alzheimer's?" 2017. http://www.alz.org/alzheimers_disease_what_is_alzheimers.asp

[30] Pdfbox, https://pdfbox.apache.org/, 2017.

Recommendation System based on CBR algorithm for the Promotion of Healthier Habits

Gineth M Cerón-Rios, Diego M Lopez-Gutierrez,
Belén Díaz-Agudo, and Juan A. Recio-García

Department of Telematic Engineering
Universidad del Cauca Popayán, Colombia
email:gceron@unicauca.edu.co, dmlopez@unicauca.edu.co
Department of Software Engineering and Artificial Intelligence
Universidad Complutense de Madrid, Spain
email: belend@ucm.es, jareciog@fdi.ucm.es

Abstract. Recommender systems are becoming very popular as they are able to predict the preferences of a user. This make recommendation based on the user profile, past ratings or/and additional knowledge such as user contextual information. Applied to the health area, they can take advantage of context information to support health promotion and disease prevention.

We present a recommender system for the promotion of physical activity called CoCARE. It recommends videos about physical activity based on a user profile, his/her context. The main challenge of CoCARE is the small set of videos to be recommended, because the selection of the videos is done manually by of health experts. Several health recommender systems have this same problem. Although today there are a large number of videos available on the Internet related to physical activity. These could not be included in the data base of CoCARE; because these do not have enough information to be categorized and profiled.

This article proposes a CBR system, this assigns a physical activity category to new video. In this way the new video will be added to the list of CoCARE recommendations. In this CBR process, basically consists on analyzing the description of the new video and compare it with the cases base of CoCARE, selecting the category of most similar cases.

1 Introduction

Recommender systems in the health area have been proven as useful tools to help patient-oriented decision making systems, promoting physical activity and disease prevention, in general, to improve health conditions through healthier habits. Health recommender systems (HRS) aim to promote health programs, to provide patients with relevant information, products or services, using knowledge about his/her personal health record systems [1].

In the literature, we find few systems that recommend health educational multimedia contents [2–6]. Users get recommendation of exercises (stretching, strengthening,etc.), with outdoor or indoor sessions, based on the user information taken from mobile devices, activity bracelets, sensors, and his/her personal health records and risk factors [7].

We have developed **"CoCARE"** a platform for promotion of healthy lifestyle on the basis of a context-aware recommendation system designed for mobile smart devices [8]. Advancements in technology, mobile devices, sensors, and wearable devices, provide users with self-monitoring dynamically acquired information of her physical activities. CoCARE recommends multimedia content of physical activity and healthy diet based on a user-context model. Given a user profile and a category, the system recommends some videos about convenient physical activities for this user at this moment. Our system relies on an initial database of activity videos that are labeled with information used during the recommendation process. Currently the system has a limited number of videos that have been manually acquired from experts in the health area.

CoCARE has a database with 80 videos. These have been tagged with its title, description, category and keywords (see example in Table 1). One video could be recommended to several users based on a decision model given by domain expert.

Title	Description	Category	Keywords
Physical Therapist Shows How To Walk Correctly	Rehab and Revive Physical Therapy We can and we will get better together! Orange County Physical Therapist and Certified Functional Manual Therapist, Dr. Lin talks about how the hip, the legs,and the arms correlate to proper walking and how it can help you walk more efficiently. Proper walking helps prevent pain and other chronic injuries.	Walk	advance, amble, foot it advance, amble, foot it.

Table 1: Example of CoCARE Videos Description

Concretely, the decision model is built from a dataset of 597 instances (rows), 6 attributes and 1 main class (see Table 2) created by the expert. CoCARE builds a decision tree using a supervised learning algorithm. Then, the system classifies the query with information about the user and his/her context and uses the tree to recommend contents based on the current user situation [9].

In this paper we deal with the problem of video acquisition and tagging. Internet provides with a huge amount of videos, most of free use, related with physical activities: dancing, running, fitness, GAP. Our goal is to use these videos as recommendation items in our system. To do that, we would need to annotate the videos with information about the potential users that would benefit from them. We propose a CBR system to automatically classify videos given its textual description. This CBR system also computes similarity between the CoCARE user profiles set and the new video categories, to found its categories.

The paper runs as follows. Section 2 describes the recommender system of CoCARE based on decision model. Section 3 explains the CBR process to automatically annotate new videos. Section 4 evaluates the CBR system. Section 6 concludes the paper and discusses some lines of future work.

2 CoCARE

CoCARE (see Figure 1) is a context aware recommender system that recommends videos on physical activity (PA) and healthy diet (HD) to patients for promotion of her healthy habits. CoCARE incorporates a context- adaptable interface based on decision trees.

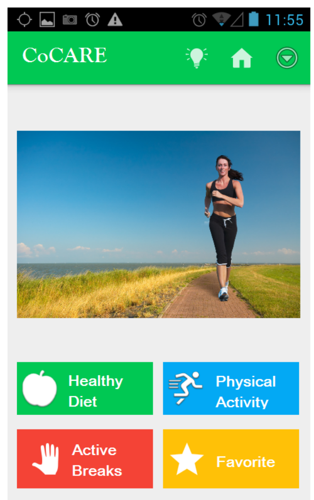


Fig. 1: Mobile CoCARE

CoCARE recommends multimedia content of physical activity and healthy diet based on user and contextual information. The basic user model includes details on the user personal profile (see Table 2). The system takes advantage of additional contextual factors to provide with personalized recommendations of multimedia content. The query includes static information like user profile, and dynamic features like geo-lacation or indoor location, date (day or season), daily schedule of the user and it can detect when the user has company. [8].

Although the CoCARE system works well as a prototype, it relies on an initial video database of 80 videos. That means different problems:

- Users get repeated contents after a while.
- Lack of novelty contents provokes user desertion.
- The task of including new videos is cumbersome.
- New videos were included without expert supervision and they were misclassified and never recommended to the right users.

We propose a CBR solution to solve these problems and assign tags (video category, keywords and user profile tags) to new videos based on the comparison to existing ones.

Attribute	Description	Value
BMI	Represents a previous inference in the user's physical condition. It is a nominal fact type.	Low weight, normal weight, overweight, obesity.
Life cycle	It is a model fact inferred from their date of birth. Despite being an integer, is taken as a nominal value in the relationships table.	Teenagers, adult.
Ethnicity	Indicates the racial group a person belongs.	It is a nominal value. Indigenous, afro, other.
Trauma	It represents a person with disability. It is a nominal fact.	Mobility, visual, auditory, without trauma.
Preference	It is the aim of the system user. It is a nominal fact.	Health, beauty, sport.
Cardiovascular disease	Indicates a user clinical condition. It is a nominal fact.	Diabetes, hypertension, without risk.
Category	It is the class of dataset. It is a nominal fact.	Dance, walk, bodily exercises, stretch, stretch eyes, limbs, personal hygiene, HIIT, labours, labours limbs, labours eyes, LISS, swim, eyes,relaxation, SCC, jog.

Table 2: User Profile Attributes

3 CBR process

Figure 2 shows the CBR process. To classify new videos, we have implemented two sequential CBR systems. The first CBR system (CBR1) receives the description of new video and returns categories from similar videos. The case base is gathered from the CoCARE video data base and contains 80 instances. Each case is described by 6 keywords and its solution is a category tag (see categories in table 2). For example:

- Description = prancing, tapping, dribbling, moving, braiding, waltz.
- Solution = dancing.
  This CBR1 module implements a k Nearest Neighbour algorithm to find the most suitable categories for a given video description. Concretely, we use a 3-NN algorithm with a keyword based similarity measure to select the three categories with highest similarity values.

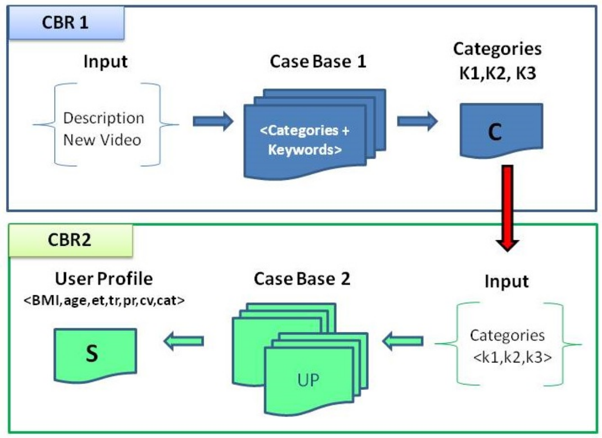


Fig. 2: CBR Process

Once the categories have been retrieved from the first case base, the second CBR module(CBR2) estimates the most suitable user profiles for the video description.

This second module has a case base with 597 instances where every case is described by several categories and 6 user profile attributes as the solution (see table 2) . By this way, the solution will be a new user profile **UP**. The algorithm compares locally the similarity value for every attribute of the user profile (see table 2) using majority voting or weighted majority voting to select the fit attribute to the solution.

$$\mathcal{CB}^1 = < C_1^1, C_2^1, \ldots, C_m^1 > \tag{1}$$
$$\mathcal{C}_i^1 = < keywords, categories > \tag{2}$$
$$\mathcal{CB}^2 = < C_1^2, C_2^2, \ldots, C_n^2 > \tag{3}$$
$$\mathcal{C}_i^2 = < categories, UP > \tag{4}$$
$$\mathcal{UP} = < BMI, age, et, tr, pr, mc, category > \tag{5}$$

4 Evaluation

We evaluate our system using leave-1-out cross-validation.

We used the video description of each case on **CB** as a query. We proposed the category and profile for UP_r, that is compared to the stored solution UP_q. We compute the similarity between attributes of UP_q and UP_r, using a binary function [0,1]. We compared if the attributes of the retrieved profile UP_r are equal to attributes of test case UP_q. So we calculated the α value (see the equation ec. 6)

$$
\begin{aligned}
Sim(UP_q, UP_r) &= \alpha \\
&\text{where} \\
a &= Sim(BMI_q, BMI_r)\epsilon[0,1] \\
b &= Sim(age_q, age_r)\epsilon[0,1] \\
c &= Sim(et_q, et_r)\epsilon[0,1] \\
d &= Sim(tr_q, tr_r)\epsilon[0,1] \\
e &= Sim(pr_q, pr_r)\epsilon[0,1] \\
f &= Sim(mc_q, mc_r)\epsilon[0,1] \\
g &= Sim(category_q, category_r)\epsilon[0,1] \\
\alpha &= 0.1 * (a + b + c + d + e + f) + 0.4 * g \qquad (6) \\
& \qquad (7)
\end{aligned}
$$

The table 3 shows an example. Our query in this example is UP_q and **D**= "Steve and Jackie take you through how to get the most out of power walking and show you how beneficial it truly is. Yes it is an Olympic sport!".

First the CBR1 module got 3NN categories as: Walk, Exercises and HIIT. So the CBR2 module compared the UP associated to these categories and found the most similarity UP_r. Next the system uses cross validation and retrieves a success solution only if the similarity measure $\alpha \geq 0.7$. This process is the comparison between attributes UP_q and UP_r. For example the table 3 shows a test case, we obtained a score of α greater than 0.7, so UP_r was added to **CB**.

Test Case	BMI	age	et	tr	pr	cv	V
UP_q	normal weight	Adult	other	without trauma	beauty	diabetes	Walk
UP_r	overweight	Young	Other	without trauma	beauty	diabetes	Walk
Test Case	a	b	c	d	e	f	g
Test Case	0	0	1	1	1	1	1

Table 3: Example cross validation

In each leave-1-out step, we obtained 3 values: the similarity of the best cases returned by the CBR1 module (1-NN), the 2 best cases (2-NN) and the 3 best cases (3-NN). Next we made 2 tests with Majority Voting (MV) and Weighted Majority Voting (WMV) in CBR2 module. Figure 3 shows results from our experiment as:

- Case 1, it is represented by the blue bar. We found the user profile with majority voting (MV) for 1NN, 2NN and 3NN.
- Case 2, it is represented by the red bar. We found the user profile with weighted majority voting (WMV) only for 2NN and 3NN.

We conclude that the better result from our CBR system was 3NN with WMV. In this case, we obtained a value of α greater than 0.85 surpassing the results achieved of the other tests. Our experiment shows that greater than 90% of the cases are correctly classified.

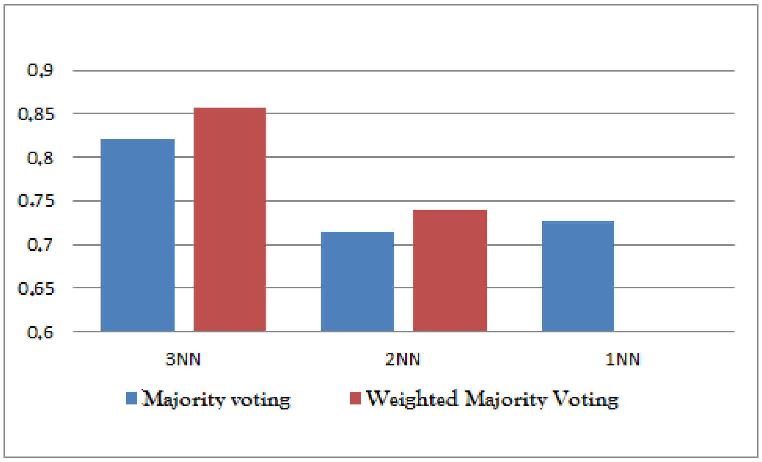


Fig. 3: Average of similarity α

5 Related works

The literature to continuation are about design and implementation of CoCARE. We approached works of health recommendation systems.

Related works such as [3, 4, 6, 10, 11] are mobile context platforms that integrates sensor technology, cognitive tutoring and evidence-based social design for health promotion. User selects group activities (jogging, walking, fitness, and yoga) to make recommendations about stretching exercises, outdoor strengthening or others, based on gender, age, weight, height, location. Diabeticlink [5], is a mobile recommender of videos and articles about exercise and healthy diabetic diet, based on user data and data of sensors. It uses the collaborative filtering recommendation technique. Finally, it generates progress reports based on the user blood glucose, his/her lifestyle, body mass index and time of physical activity. Kalico [12] is a mobile recommender system of healthy food restaurants, the user suggestion are based in his/her profile, location, budget and preferences. It provides a list of nearby restaurants in alphabetical order and presents a list of healthy menus recommendation in each it. Kalico is a system that only promotes healthy eating and for people who want to eating out.

The previously works mention the use of user models, data modeling and use of recommendation techniques, but they do not describe the selection of recommendation techniques and how performance the validation them. On the other hand, these works didn‘t make mention about how many recommendations have their systems, apparently the systems has few contents to recommender.

The next related works relate with CBR topic. [13, 14], these works use a CBR algorithm to recommend diabetes care videos to adults, however there is no evidence that their systems can retrieves additional information from the videos description. There are other systems that retrieve textual information [10, 15, 16], recover the sentences that are necessary and complete the sentence, others recover symptoms of some disease with the user profile. However, there is no evidence a system that finds user profiles with just the description of an item (video).

Our CBR systems retrieves a user profile from the video description. This could be useful in others areas such as education, commerce and / or advertising. For example one recommendation systems could find a user profile fit for learning content or advertising videos with just the description of item.

6 Discussion and Conclusions

We have described our CoCARE recommender system. CoCARE recommends videos of physical activity categorized by health experts. But the problem is that they are very few, to include a new video must be properly categorized for a user profile. Our CBR allows you to categorize the video and find an appropriate profile from the description of a video. In this work we proposed a system composed of 2 CBR system, the first categorizes the new video and the second delivers the appropriate profile. We have evaluated that the CBR system delivers a better response if the first CBR is 3NN and CBR2 is with similarity.

Our CBR system uses little input knowledge to get an adequate solution. It offers a simpler alternative to associate videos to the needs and preferences of different users.

Our system benefits the user and the health expert, with the possibility of having new recommendations that help the adherence of the physical activity program.

The work presented in this paper opens several lines of future work.

When you have very short video descriptions the CBR system loses precision in finding the right category, although the results we obtained are very promising we have considered that they can be improved if we extend the description from synonyms using an ontology of synonyms and algorithms matching of learning.

We plan to take information about the most viewed videos on the Internet (YouTube) and use their description to classify them, assign to the new video an appropriate user profile and add to CoCARE case base **CB** automatically using collaborative filtering and CBR.

7 Acknowledgments

This work was performed under the doctoral thesis “Context-Aware Recommender System to Physical Activity Promotion” financed by Colciencias, under call “Convocatoria No 6172 (Doctorados Nacionales)”. Supported by UCM (Group 921330) and Spanish Committee of Economy and Competitiveness (TIN2014-55006-R).

References

1. Wiesner, M., Pfeifer, D.: Health recommender systems: concepts, requirements, technical basics and challenges. International journal of environmental research and public health **11**(3) (2014) 2580–2607
2. Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-Aware Recommender Systems. AI MAGAZINE **32**(4) (2011) 217–253
3. F, B., L., C.: Mopet: a context-aware and user-adaptive wearable system for fitness training.. Artificial Intelligence in Medicine **42**(2) (2008) 153–163
4. Lamprinos, I., Papadaki, C.: Mobile personal health application for empowering diabetic patients: The case within empower project. In: Proceedings of The International eHealth Telemedicine and Health ICT Forum. Luxembourgh. (2013)
5. Chen, H., Compton, S., Hsiao, O.: Diabeticlink: A health big data system for patient empowerment and personalized healthcare. In: Smart Health, Springer (2013) 71–83
6. Du, H., Youngblood, G.M., Pirolli, P.: Efficacy of a smartphone system to support groups in behavior change programs. In: WH ’14 Proceedings of the Wireless Health 2014 on National Institutes of Health, ACM (2014) 1–8
7. Mettouris, C., Papadopoulos, G.A.: Contextual Modelling in Context-Aware Recommender Systems: A Generic Approach. Web Information Systems Engineering WISE 2011 (2013) 41–53
8. González, K.X., Carvajal, M.Á., Cerón, G.M., López, D.M.: Modelado de un sistema consciente del contexto para soportar intervenciones en actividad física y nutrición saludable. Revista Ingenierías Universidad de Medellín **15**(28) (2016)
9. Girón, E., Cerón, G.M., Rico, C., López, D.M.: Framework for data model to personalized health systems. In: 2016 8th Euro American Conference on Telematics and Information Systems (EATIS). (2016) 1–5
10. El-Sappagh, S.H., El-Masri, S., Elmogy, M., Riad, A.M., Saddik, B.: An ontological case base engineering methodology for diabetes management. Journal of Medical Systems **38**(8) (2014) 67
11. Cochrane, K., Schiphorst, T.: Developing design considerations for mobile and wearable technology: M-health applications that can support recovery in mental health disorders. In: Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare. PervasiveHealth ’15, ICST, Brussels, Belgium, Belgium, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2015) 29–36
12. Anwar, M., Hill, E., Skujins, J., Huynh, K., Doss, C.: Kalico: A smartphone application for health-smart menu selection within a budget. In: International Conference on Smart Health, Springer (2013) 113–121

13. Bermingham, A., O'Rourke, J., Gurrin, C., Collins, R., Irving, K., Smeaton, A.F.: Automatically recommending multimedia content for use in group reminiscence therap. In: Proceedings of the 1st ACM International Workshop on Multimedia Indexing and Information Retrieval for Healthcare. MIIRH '13, New York, NY, USA, ACM (2013) 49–58
14. Polignano, M., de Gemmis, M.: Recommender systems supporting decision making through analysis of user emotions and personality. In: DC@ AI* IA. (2015) 43–48
15. Linckels, S., Meinel, C.: Applications of description logics to improve multimedia information retrieval for efficient educational tools. In: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval. MIR '08, New York, NY, USA, ACM (2008) 321–328
16. Sandhu, R., Kaur, J., Thapar, V.: An effective framework for finding similar cases of dengue from audio and text data using domain thesaurus and case base reasoning. Enterprise Information Systems **0**(0) (0) 1–18

Preface

The Doctoral Consortium proceedings contains the research summaries that were presented at the 9th Annual ICCBR 2017 Doctoral Consortium which was held on Monday June 26th 2017 in Trondheim, Norway. There were 9 accepted submissions consisting of (i) an application cover page, (ii) a research summary, (iii) a curriculum vitae and (iv) a letter of support from the student's advisor.

The objectives, progress, plans and references in each research summary were progressively refined according to feedback from two PC members. Feedback was organised into three broad areas: general outlook in terms of research hypothesis and proposed methodology; detailed comments specific to the student's project; and finally advice for the talk presentation.

Participants in the Doctoral Consortium were assigned a mentor. A face-to-face pre-event meeting opportunity, held on June 25th, enabled all student-mentor pairs to meet in person, and to refine their presentations. The evening ended with the DC participants and mentors meeting with other conference participants for dinner.

On June 26th, the formal program started with an invited talk by Dr Odd-Erik Gundersen from NTNU. The next sessions featured 20-minute talks presented by the nine doctoral students on their research summary. Mentors had the responsibility of leading the question and answer session following each mentee presentation. A final wrap-up session concluded the day. The presentations covered a wide range of CBR topics including similarity and retrieval, process-oriented CBR, case-based maintenance, CBR and big data. Healthcare as well as industrial applications were described.

Many people participated in making the DC event a success. We wish to thank all our PC members who provided important and useful guidance to DC students, either as reviewers or as mentors. We are very grateful for the generous support of the National Science Foundation which helped fund travel costs for our students from the US.

Finally thank you to all our DC participants. We had a returning PhD student participant which was a valuable indicator that the DC at ICCBR is a useful and beneficial event. We trust that the ICCBR-17 DC enhanced your interest in studying CBR and that the welcome and support from the CBR community has reinforced your interest in this field for the future.

June, 2017

Stefania Montani
Jonathan Rubin

Program Chairs

Stefania Montani	University of Piemonte Orientale, Italy
Jonathan Rubin	Philips Research North America

Program Committee

Agnar Aamodt	NTNU, Norway
Ralph Bergmann	University of Trier, Germany
Isabelle Bichinaritz	State University of New York at Oswego, USA
Alexandra Coman	NRC/Naval Research Laboratory, USA
Klaus-Dieter Kalthoff	DFKI/University of Hildesheim, Germany
Michael Floyd	Knexus Research, USA
Stelios Kapetanakis	University of Brighton, USA
Luc Lamontagne	Laval University, Canada
David Leake	Indiana University, USA
Cindy Marling	Ohio University, USA
Luigi Portinale	University of Piemonte Orientale, Italy

A framework for multi-level semantic trace abstraction

Doctoral Consortium ICCBR 2017

Manuel Striani

PhD Candidate
Department of Computer Science, University of Torino
Corso Svizzera 185, 10149 Torino, Italy
striani@di.unito.it

1 Research Summary

Many commercial information systems and enterprise resource planning tools routinely adopted by organizations and companies worldwide, like those provided by, e.g., Oracle and SAP, record information about the executed business process instances in the form of an *event log* [5]. The event log stores the sequences (*traces* henceforth [3]) of actions that have been executed at the organization, typically together with key execution parameters, such as times, costs and resources.

Event logs constitute a very rich source of information for several business process management tasks. Indeed, the experiential knowledge embedded in traces is directly resorted to, e.g., in *operational support* and in *agile workflow* tools, which can take advantage of **trace comparison** and **retrieval**. Operational support [3] assists users while process instances are being executed, by making predictions about the instance completion, or recommending suitable actions, resources or routing decisions, on the basis of the comparison to already completed instances retrieved from the log. The agile workflow technology [10, 8] deals with adaptation and overriding needs in response to expected situations (e.g., new laws, reengineering efforts) as well as to unanticipated exceptions and problems in the operating environment (e.g., emergencies) [4], even if the default process schema is already in use by some running instances [9, 2]: in order to provide an effective and quick adaptation support, many agile workflow systems share the idea of recalling and reusing concrete examples of changes adopted in the past, recorded as traces in the event log. The CBR [1] methodology, and in particular the retrieval step, can therefore be adopted in this context.

In my PhD thesis, I am developing a framework to compare and retrieve process traces, represented at **different levels of abstraction**. The framework will then be interfaced to operational support or agile workflow tools, as well as to other analysis mechanisms. In this paper, I describe the methodological approach behind trace abstraction; the applications mentioned above will be considered in my future work.

1.1 Multi-level abstraction mechanism

We are developing a **semantic-based, multi-level abstraction mechanism**, able to operate on event log traces. In our approach, actions in the log are mapped to instances of ground concepts (leaves) in a taxonomy, so that they can be converted into higher-level concepts by navigating the hierarchy, up to the desired level, on the basis of the user needs.

The **abstraction mechanism** has been designed to properly tackle non-trivial issues that could emerge. Specifically:

- two actions having the same ancestor in the taxonomy (at the chosen abstraction level) may be separated in the trace by a *delay* (i.e., a time interval where no action takes place), or by actions that descend from a different ancestor (*interleaved actions* henceforth). Our approach allows to deal with these situations, by creating a single *macro-action*, i.e., an abstract action that covers the whole time span of the two actions at hand, and is labeled as the common ancestor; the macro-action is however built only if the total delay length, or the total number/length of interleaved actions, do not overcome proper admissibility thresholds set by the user. The delays and interleaved actions are quantified and recorded, for possible use in further analyses. In particular, we have defined a **similarity metric** where this information is accounted for as a penalty, and affects the similarity value in abstract trace comparison;
- abstraction may generate different types of temporal constraints between pairs of macro-actions; specifically, given the possible presence of interleaved actions, we can obtain an abstracted trace with two (or more) overlapping or concurrent macro-actions. Our approach allows to represent (and exploit) this information, by properly maintaining both quantitative and qualitative temporal constraints in abstracted traces. Once again, this temporal information can be exploited in further analyses. In particular, the **similarity metric** we adopt in trace comparison can deal with all types of temporal constraints.

Specifically, the procedure to abstract a trace operates as follows:

- for every action i in the trace:
  - i is abstracted as its ancestor at the taxonomy level selected by the user; the macro-action m_i, labeled as the identified ancestor, is created;
  - for every element j following i in the trace:
    - if j is a delay, its length is added to a variable $tot - delay$, that stores the total delay duration accumulated so far during the creation of m_i;
    - if j is an interleaved action, its length is added to a variable $tot - inter$, that stores the total interleaved actions durations accumulated so far during the creation of m_i;
    - if j is an action that, according to domain knowledge, abstracts as the same ancestor as i, m_i is extended to include j, provided that

$tot - delay$ and $tot - inter$ do not exceed domain-defined thresholds. j is then removed from the actions in the trace that could start a new macro-action, since it has already been incorporated into an existing one;

- the macro-action m_i is appended to the output abstracted trace which, in the end, will contain the list of all the macro-actions that have been created by the procedure.

The variables $tot - delay$ and $tot - inter$, accumulated during abstraction, are also provided as an output attribute of each macro-action and they will be used as a penalty in abstracted trace similarity calculation.

The most significant and original methodological contributions of the work thus consist in:

1. having defined a proper ***mechanism for abstracting event log traces***, able to manage non trivial situations (originating from the treatment of interleaving actions or delays between two actions sharing the same ancestor);
2. having provided ***a trace comparison facility***, which resorts to a ***similarity metric*** (extending the metric presented in [6]), able to take into account also the information recorded during the abstraction phase.

In the third year, I will concentrate on experimental work referring to trace comparison and I will deal with operational support, agile workflow management, or other activities, including process mining on abstracted traces. As regards process mining, in particular, we wish to test when abstraction allows to make clear and more readable process model.

1.2 Current development stage

With the help of an expert physician in stroke patient management, we have formalized medical domain knowledge in a taxonomy (which has been organized by goals) by using the Protègè ontology editor [7]. Actions in traces are mapped to the taxonomy leaves, so navigating the taxonomy it is possibile to abstract actions by goals. we have worked on a metric for trace comparison that is able to manage both temporal and non temporal information in traces, and to take into account information collected during the abstraction process.

The system architecture we have developed, is shown in Figure 1. Rectangles represent computational modules, while ovals and cylinders represent domain knowledge sources and the database. The first step to be executed is *event log preparation*, that takes in input the available database (DB), and exploits domain knowledge (the taxonomy); the event log will then undergo *abstraction*. The abstracted event log will be given as an input to *trace comparison* resorting to the metric we have developed, or to *process mining*, *operational support*, or other activities, that we plan to realize by resorting to ProM.

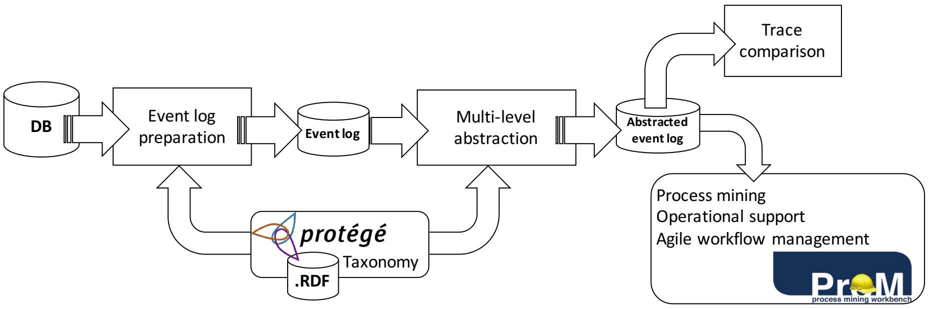


Fig. 1. Framework architecture and data flow

1.3 Future work

During my last PhD year, the framework will be tested in the field of stroke management, where we will adopt multi-level abstraction and trace comparison to cluster event logs of different stroke units, in order to highlight correct and incorrect behaviors, abstracting from details (such as local resource constraints or local protocols). The goal will be to show that, the application of the abstraction mechanism allows to obtain more homogeneous and compact clusters (i.e., able to aggregate closer examples), still making outliers clearly identifiable, and isolated in the cluster hierarchy. Some first encouraging results are already available.

As regards process mining, the ground processes (process learned on trace at the same level of taxonomy leaves) are typically "spaghetti-like": they presents an extremely large number of nodes and edges which make it hard to identify details. Our hypothesis is that models learned on abstracted traces will be much more compact and it will be possible for medical experts to analyze them. This topic will be studied during my last year as well.

Finally, we will provide abstracted traces as an input to operational support or agile workflow management facilities.

References

1. A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations and systems approaches. *AI Communications*, 7:39–59, 1994.
2. F. Casati, S. Ceri, B. Pernici, and G. Pozzi. Workflow evolutions. *Data and Knowledge Engineering*, 24:211–238, 1998.
3. W. Van der Aalst. *Process Mining. Discovery, Conformance and Enhancement of Business Processes.* Springer, 2011.
4. P. Heimann, G. Joeris, C. Krapp, and B. Westfechtel. Dynamite: dynamic task nets for software process management. In *Proceedings International Conference of Software Engineering*, pages 331–341, Berlin, 1996.
5. $http://www.win.tue.nl/ieeetfpm$. IEEE Taskforce on Process Mining: Process Mining Manifesto (last accessed on 4/11/2013).
6. S. Montani and G. Leonardi. Retrieval and clustering for supporting business process adjustment and analysis. *Information Systems*, 40:128–141, 2014.
7. The Protégé Team. The Protégé website: urlhttp://protege.stanford.edu/, 2013.
8. M. Reichert and P. Dadam. Adeptflex-supporting dynamic changes of workflows without losing control. *J. Intell. Inf. Syst.*, 10:93–129, 1998.
9. S. Rinderle, M. Reichtert, and P. Dadam. Correctness criteria for dynamic changes in workflow systems - a survey. *Data and Knowledge Engineering*, 50:9–34, 2004.
10. B. Weber and W. Wild. Towards the agile management of business processes. In K. D. Althoff, A. Dengel, R. Bergmann, M. Nick, and T. Roth-Berghofer, editors, *Professional knowledge management WM 2005, LNCS 3782*, pages 409–419, Washington DC, 2005. Springer, Berlin.

A Case-Based Real-Time Adaptive Engineer Site Support System

Kyle Martin

School of Computing Science and Digital Media,
Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
k.martin@rgu.ac.uk

Abstract: Employee experience is a valuable asset for any company. A system which can store, retrieve and adapt these experiences to meet the requirements of new scenarios can play an important role in corporate knowledge management. Case-Based Reasoning provides an excellent mechanism for this because it allows the capture and reuse of past experiences, which in this paper involves the generation of expert-level support in response to questions raised by British Telecommunications field engineers. However, experience capture is hard in dynamic environments involving multi-modal communication and content. This paper examines the context of this research and details the work which has been completed thus far, as well as potential next steps.

Keywords: Case-Based Reasoning · Siamese Neural Networks · Knowledge Management

1 Introduction

This research project is a collaboration between British Telecommunications (BT) and Robert Gordon University to produce a system which uses experiential content gathered from users as a case-base to answer new queries. The query and answer process will take place while engineers are out in the field, giving them access to the support they require within the dynamic environment of their jobs. This would facilitate the exchange of knowledge and experience between employees within the company and assist in the development of a 'corporate memory', which stores the relevant experiences of every engineer in the company, preventing these assets from being lost if an employee were to leave BT.

This paper is structured as follows: section 2 gives an overview of the context of the project and section 3 describes its contributions. Section 4 discusses related work and research. The report concludes with a description of the research which has been completed so far and details future work and next steps the research could take.

2 An Overview of the Project

The goals of this project are to develop a means of capturing experiential knowledge from BT engineers and to produce an application which can learn to provide expert-level support in the field using that information. The system should be able to support engineers by retrieving relevant results and adapting to new situations accordingly, producing results which are applicable in the real-world.

In the field, an engineer is only able to access notes (historical task records) which are part of the current task. These notes are plain text and contain customer and location information, but rarely indicate what piece of equipment the fault may relate to (as this may not be known at the time of task allocation), the context of the task, or similar tasks. An engineer is expected to rely heavily on past experiences and training in order to diagnose and solve problems, but this can fail if the task has novel, unusual or specialist elements. Some of these failures may be avoided if there existed a means of drawing upon the experiences of engineers who had previously encountered similar problems.

To give a real example, a BT power engineer was called out to a 'low-voltage' alarm in a rural exchange. It transpired that the mains power had failed and the back-up generator had not started, so the exchange was beginning to lose power and risked the network going down for nearby customers. The engineer was unsure of what the exact fault with the back-up generator could be and spent some time attempting to diagnose the problem before calling for help. It was only when an engineer with more experience arrived that the fault was eventually diagnosed - the ac/mc contactor within the generator needed replaced. This was a time-dependent and critical fault which could have had important business repercussions and was only solved because a more experienced engineer was able to attend on short notice. If experience and knowledge could be more effectively transferred between engineers, then we could severly reduce the time taken to solve these faults, as well as the manpower required to do so and the risk of failing to complete time-critical faults.

We aim to use the notes and other knowledge assets (including video, photo and audio content) as the case base for an intelligent system which can reply to engineer's queries and propose a solution. This would alleviate pressure upon engineers by giving them access to a 'digital expert' which could draw upon historical experiences from the entire work force to provide support in the field.

3 Contributions

As this project is a collaboration between a representative of industry and a university, it is important that the research provides an academic contribution, but remains viable for use in a commercial setting. There often exists a disconnect between the two, fuelled by the exploratory nature of research and the fact that 'state-of-the-art' measures may involve expensive procedures or equipment which do not have commercial viability. Therefore, this project is a case study of a research project which has both academic importance and good business viability and offers contributions in both areas.

The main contribution of this project towards industry will be to improve information access for engineers in the field and facilitate the exchange of knowledge and expertise by using an intelligent system. By doing so, this system should ultimately improve an engineer's ablity to complete a task and, at an organisational level, increase overall engineer productivity.

The contribution of this project to academia is to develop a dynamic decision support system which can operate on a large scale across multiple engineer domains. This project will ultimately showcase a system which can retrieve relevant results from vast quantities of complex, inter-related multi-media data.

4 Related Works

Due to its very nature of reusing and adapting previous examples, Case-Based Reasoning (CBR) is the branch of machine learning which most closely reflects the goals of the project. Using CBR techniques to facilitate knowledge flow between users is not a new concept, having been pursued with differing levels of success for a number of years. Of these, many projects have specifically targeted domain experts within a pre-identified industry niche, including food quality control [8] and help desk support [5], in order to provide relevant assistance based upon experts' and users' experiences.

In [6] Goker et al developed an adaptive expertise provider dubbed the Pricewaterhouse Cooper (PwC) Connection Machine. The Connection Machine allowed users to enter their queries into a web application and made use of CBR techniques in order to identify experts who may be able to answer. Use of the system facilitated the exchange of knowledge and experience between users and provided a singular forum for accessing all experts within the company. The biggest disadvantage of the system was that it relied upon experts to actively answer queries. Drawing upon this idea, our project aims to allow users to access the sum of all experience of BT engineers in a single place, but remove the need for human experts to explicitly answer queries. The system will return relevant answers based upon its knowledge gained from the input task notes.

A Case Retrieval Net (CRN) is a CBR framework which facilitates the return of a small number of cases in a large case base. CRNs use a memory structure that stores both the contents of the case base and similarity knowledge between cases [9] using Information Entities (IEs). An IE is any specific piece of information pertaining to a case (such as an attribute-value pair). Results are returned using 'spreading activation'; the most relevant IE to a query is activated and nearby IEs receive diminishing activation the less similar they are to the identified IE. The case nodes associated with the activated IEs are then collected and returned. CRNs have demonstrated promising results in reuse of textual cases within large medical databases [1]. The return of a small number of relevant results from a massive case base and the use of similarity knowledge to facilitate case adaptation are both vital components of the project. However, this requires a method of generating the extensive similarity knowledge required for spreading activation of the net. This may be achieved using the object-to-object similarity generated by a Siamese Neural Network.

A Siamese Neural Network (SNN) architecture consists of two neural networks that share identical weights and are joined at one or more layers. Introduced in [2] as a method of signature verification, SNNs are trained and tested on pairs of examples to develop similarity knowledge at a case-to-case level. Desirable pairs are dubbed as 'genuine' during training, while undesirable pairs are 'impostors', so that the network develops vectors representative of case features. At test time, the SNN measures the distance between the queried vectors to determine whether they are 'genuine' or 'impostor' based on a threshold.

Recent research has demonstrated that SNNs are able to generate object-to-object similarities after being trained with relatively few examples or in datasets where a vast number of classes exist [7]. This could be particularly useful in the current project, where the broad domain could mean that there are a massive number of classes within the case base. SNNs have been applied with success in areas like sketch-based retrieval [10], and speaker recognition [3]. In [4], an SNN is applied to the task of similar question retrieval, and outperforms the state-of-the-art. In the same way, this project would aim to return similar cases to the situation described in the query, but unlike in the examples above it would also attempt to adapt these cases to better suit the described situation.

5 Current and Future Work

Much of our recent work has been gathering data to determine the industrial and academic context of this project. In particular, we reviewed literature featuring industry examples of experience capture and knowledge transfer systems to see how others have dealt with similar problems. In addition, we gathered data from within BT to establish the specific business context of the project. This involved determining the available information sources for use by engineers, how they are used on specifc tasks and in what areas they are lacking.

One of the key aspects of this project is the development of a large and dynamic case base which can be used and updated in real-time throughout the day. Often, retrieval from a huge case base can be extremely costly. We are examining methods of reducing this cost without sacrificing case base coverage or retrieval accuracy through similarity-based retrieval in a CRN, but learning similarity knowledge in a huge system can be expensive. To this end we are performing experiments to learn similarity between cases in a quick and inexpensive manner. Recently we have examined generating case-to-case similarity knowledge by using an SNN and have demonstrated that this is capable of developing case-to-case similarity knowledge suitable for similarity-based retrieval.

In future work, we would like to experiment with training SNN on limited data to ascertain whether they can successfully learn similarity knowledge. Also, we would like to examine populating the IEs of a CRN using the values developed by the output of an SNN to see whether we can return improved results with spreading activation.

References

[1] Adeyanju, I. et al. 2009. Case Retrieval Reuse Net (CR2N): An Architecture for Reuse of Textual Soutions. In: L. McGinty, D. C. Wilson (eds) Case-Based Reasoning Research and Development. ICCBR 2009. Lecture Notes in Computer Science, vol. 5650. Berlin: Springer-Verlag

[2] Bromley, J. et al. 1993. Signature Verification Using a"Siamese" Time Delay Neural Network. International Journal of Pattern Recognition and Artificial Intelligence, 7(4) pp. 669 - 688.

[3] Chen, K and Salman, A. 2011. Extracting Speaker-Specific Information with a Regularized Siamese Deep Network. In: J. Shawe-Taylor et al (eds) Advances in Neural Information Processing Systems 24 (NIPPS 2011). Granad, Spain, December 12 17, 2011.

[4] Das, A. et al. 2016. Together We Stand: Siamese Networks for Similar Question Retrieval. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016). Berlin, Germany, August 7 - 12, 2016. Stroudsburg, PA, USA: ACL

[5] Goker, M.H. and Roth-Berghofer, T. 1999. Development and Utilization of a Case-Based Help-Desk Support System in a Corporate Environment. In: K-D. Altho et al. (eds) Case- Based Reasoning Research and Development: Third International Conference on Case-Based Reasoning, ICCBR-99. Seeon Monastery, Germany. July 27 - 30. Berlin: Springer-Verlag

[6] Goker, M.H. et al. 2006. The PwC Connection Machine: An Adaptive Expertise Provider. IN: Thomas R. Roth-Berghofer et al. (eds) Advances in Case Based Reasoning 8th European Conference, ECCBR 2006. Fethiye, Turkey, September 4 - 7, 2006. Berlin: Springer-Verlag.

[7] Koch, G. et al. 2015. Siamese Neural Networks for One-Shot Learning. In: ICML 2015. Lille Grand Palais, Lille, 6 11 July, 2015.

[8] Lao, S. I. et al. 2012. Achieving Quality Assurance Functionality in the Food Industry Using a Hybrid Case Based Reasoning and Fuzzy Logic Approach. Expert Systems with Applications. Vol 39. pp 5251 - 5261

[9] Lenz, M. and Burkhard, H. D. 1996. Case Retrieval Nets: Foundtions, Properties, Implementation and Results. [Technical Report] Berlin, Germany: Humboldt University

[10] Qi, Y. et al. 2016. Sketch-Based Image Retrieval via Siamese Convolutional Neural Network. In: 2016 IEEE ICIP. Phoenix Convention Centre, Phoenix, USA. September 25 28, 2016. Red Hook, NY, USA: Curran Associates

Building an Integrated CBR-Big Data Oriented Architecture for Case-Based Reasoning Systems

Kareem Amin[1,2]

[1] PhD Candidate, German Research Center for Artificial Intelligence, Smart Data and Knowledge Services, Trippstadter Strae 122, 67663 Kaiserslautern, Germany,
[2] Big Data Consultant, Sulzer GmbH, Frankfurter Ring 162, 80807, Munich, Germany
kareem.amin@dfki.uni-kl.de,
kareem.amin@sulzer.de

1 Introduction

The growth of intensive data-driven decision-making is now being recognized broadly. Big data systems are mainstream and the demand for building systems that able to process data streams is growing. Yet many decision support systems act like "black boxes", providing little or no transparency in the rationale of their processes [1]. The "black box" methodologies are not acceptable in crucial domains like health care, aviation, and maintenance. Experts prefer to reason the decisions. Current big data strategies tend to process in-motion data and offer many potential scenarios to work with. The big data term refers to dynamic, large, structured and unstructured volumes of data generated from different sources with different formats [2]. Therefore, it is a must for CBR systems that tends to process the in-motion data to manage their sub-tasks, such as collecting and formatting data, case base maintenance, cases retrieval, cases adaptation and retaining new cases [3]. In my research I will describe the idea of spanning the gap between CBR and Big Data based on the SEASALT architecture [4] [5]. SEASALT is an application independent architecture to work with heterogeneous data repositories and modularizing knowledge. It was proposed based on the CoMES approach to develop collaborative multi-expert systems and provides an application-independent architecture that features knowledge acquisition from a Web community, knowledge modularization, and agent-based knowledge maintenance. Its first research prototype was developed for the travel medicine application [4]. SEASALT aims to provide a coherent multi-agent CBR architecture that can define the outlines and interactions to develop multi-agent CBR systems.

2 CBR & Big Data

When CBR research has addressed increased data sizes, the primary focus has been compression of existing data rather than scale-up. Considerable CBR research has focused on the efficiency issues arising from case-base growth. As the case base grows, the swamping utility problem can adversely affect case retrieval times, degrading system performance [6]. CBR and Big Data collaboration is an emerging topic, some researches have been carried out focusing mainly on case base maintenance methods, aiming to reduce the case base size while preserving competence [8][7]. Few CBR projects have considered scales up to a million of cases [10][9]. The ability of case-based reasoning to reason from individual examples and its inertia-free learning makes it appear a natural approach to be applied to big-data problems such as predicting from very large example sets [6]. Likewise, if CBR systems had the capability to handle very large data sets, such a capability could facilitate CBR research on very large data sources already identified as interesting to CBR, such as cases harvested from the Experience Web [11], cases resulting from large-scale real-time capture of case data from instrumented systems [12], or cases arising from case capture in trace-based reasoning [13].

3 Research Focus

In my thesis I am going to concentrate on building a multi-agent CBR system that extends the SEASALT architecture. The proposed approach is designed to semi-automate the building of cases based on chunks of data coming from different streams, and being able to work with big number of historical cases stored in our case base. A real use case to elaborate the main goal of my model would be in manufacturing [14]. In manufacturing processes data comes from different machines and sensors. We need to detect any pattern that has led to a disqualified end product, and give a proactive solution to avoid or mitigate the effect of these kinds of patterns [15]. Hence, from the Big Data 4V's, I will mainly focus on velocity and volume with lower exposure to variety. I need to collect data from different sources and be able to detect patterns that match our old cases in real time. To achieve the aforementioned goals, the following objectives have to be fulfilled:

1. Extend the original SEASALT architecture with a new layer "Knowledge Stream Management"
2. Correlate and synchronize between the chunks of data that come from different sources
3. Collect sufficient knowledge from domain experts that help in achieving point 2
4. Develop a methodology to apply the new approach to existing multi-agent systems as well as integrating it into the development of new multi-agent systems
5. Evaluate the new approach and the methodology within an industrial use case
6. Compare performance and accuracy with other existing techniques and systems

The proposed approach is roughly described in details in the following sections.

4 The Knowledge Stream Management

The original idea of the SEASALT architecture comes from Althoff, Bach and Reichle [5]. SEASALT consists mainly of five main layers, Knowledge Source, Knowledge Formalization, Knowledge Representation, Knowledge Provision, and Individualized Knowledge. Every layer contains several software agents designated for several tasks. Through my work, a new layer will be added: "Knowledge Stream Management" (See Figure 1). The new layer has two main tasks, the first is processing the streams of data coming from Knowledge Sources in real time, and the second is to give real time analysis to data patterns found within the streams. The Knowledge Stream Management layer will contain software agents designated for the prescribed tasks. System nodes [3] would be the available processing power. The Knowledge Provision layer will be distributed across several nodes, and hence each node contains Knowledge Provision agents. The Coordination Agent will act as the system manager who is aware of all the system nodes and responsible for the whole system control. He will be the data tap that uses the underlying framework to distribute the incoming requests across the system nodes. Normally, there are two kinds of nodes, one for Queries processing to retrieve results and the second for New Cases processing. It is possible to have up to N nodes in the system according to the volume of data that should be processed in real time. According to Big Data system architecture and sizing best practices provided from Hortonworks [4], for sustained throughput of 50MB/sec and thousands of events per second, we need 1-2 nodes and 8+ cores per node (more is better), 6+ disks per node (SSD or Spinning) and 2 GB of memory per node and 1GB bonded. In every node, there would be a Classification Agent to classify the received data chunks and assign it to the intended Topic Agent. Each Classification Agent is aware of the knowledge map gathered from knowledge sources and classify the incoming data according to predefined classes (collected before from domain experts). Then, the Classification Agent assigns the request to the intended Topic Agent(s). The Topic Agent is performing queries to retrieve the most similar cases. Since distributed nodes are being used in the hardware cluster, the Case Base will be replicated to avoid data integrity problems using replication channels to replicate data between all Case Base instances. The Case Factory agents will be centralized, and hence the Case Factory will have only one instance that performs case maintenance on a single Case Base. Afterwards, the results will be distributed to the whole system nodes using the replication channels.

We assume that solving big data problems will require also manual knowledge modelling. CBR - standing with one foot in the area of Machine Learning [automated knowledge generation] and with the other foot in the area of Knowledge-Based Systems [manual and semi-automatic knowledge modelling] -

[3] Every single node is a processing power

[4] Hortonworks is one of the biggest big data software companies founded in June 2011 as an independent company based in Santa Clara, California. http://www.hortonworks.com

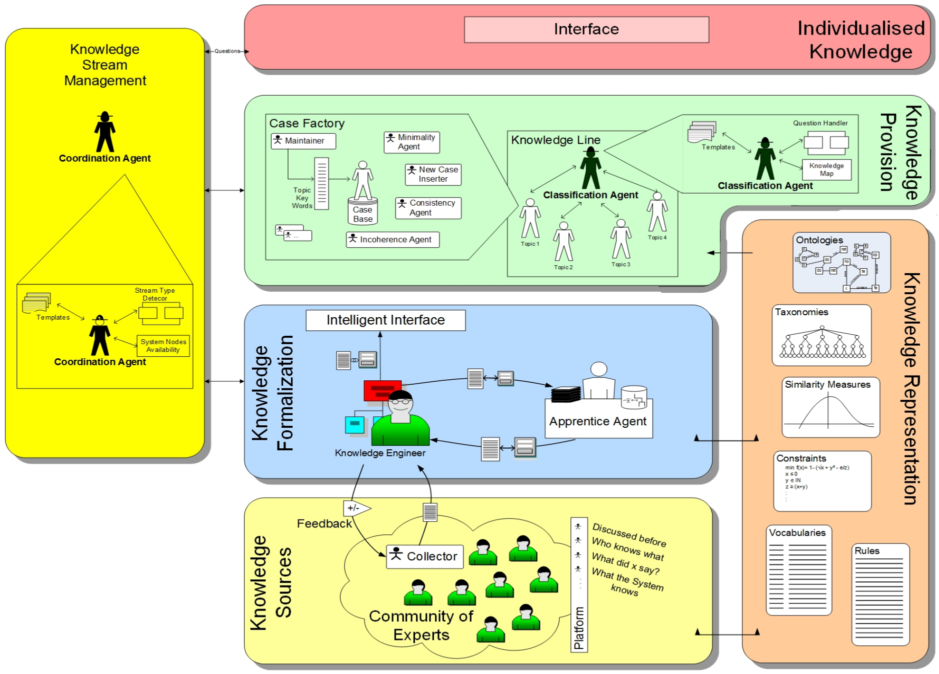


Fig. 1. Big Data Oriented SEASALT Architecture

is a natural candidate for finding a domain-and task-specific approach of integrating automated knowledge generation [using machine learning] with manual knowledge modeling [using knowledge-intensive CBR].

4.1 Potential Applications

1. Internet of Things (IoT) applications.
2. Ticketing systems and customer support applications.
3. Server logs anomaly detection applications.
4. Condition monitoring applications.

5 Current Progress & Future Directions

Currently I am shaping my PhD goals and approach. I intend to implement our approach and compare accuracy and speed performance with other case base maintenance methods. I am currently working to learn the big data system architectures and tools, that will help in the implementation phase. In the meanwhile, I am trying to find a suitable industrial use case to apply the proposed approach.

References

1. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining Collaborative Filtering Recommendations. In: Proceedings of the 2000 ACM Conference on Computer supported cooperative work, ACM (2000).
2. A community white paper developed by leading researchers across the United States, "Challenges and Opportunities with Big Data," Purdue University, USA, 2012.
3. Aitor Mata, "A Survey of Distributed and Data Intensive CBR Systems," SpringerVerlag Berlin Heidelberg, pp. 582-586, 2009.
4. Meike Reichle, Kerstin Bach and Klaus-Dieter. Althoff, "Knowledge engineering within the application-independent architecture SEASALT" International Journal Knowledge Engineering and Data Mining, vol. 1.1, no. 3, pp. 202-215, 2011.
5. Kerstin Bach,Meike Reichle and Klaus-Dieter. Althoff, "A Domain Independent System Architecture for Sharing Experience," Proceedings of LWA 2007, Workshop Wissens- und Erfahrungsmanagement, September 2007, pp. 296-303
6. Vahid Jalali and David Leake, "CBR Meets Big Data: A Case Study of Large-Scale Adaptation Rule Generation," Case-Based Reasoning Research and Development, pp. 181-196, 2015.
7. Smyth, B., Keane, M.: Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, San Mateo, Morgan Kaufmann (1995) 377382
8. Smyth, B., McKenna, E.: Building compact competent case-bases. In: Proceedings of the Third International Conference on Case-Based Reasoning, Berlin, Springer Verlag (1999),329342
9. Daengdej, J., Lukose, D., Tsui, E., Beinat, P., Prophet, L.: Dynamically creating indices for two million cases: A real world problem. In: Advances in Case-Based Reasoning, Berlin, Springer (1996) 105119
10. Beaver, I., Dumoulin, J.: Applying mapreduce to learning user preferences in near realtime. In: Case-Based Reasoning Research and Development, ICCBR 2014, Berlin, Springer (2014) 1528
11. Plaza, E.: Semantics and experience in the future web. In: Proceedings of the Ninth European Conference on Case-Based Reasoning, Springer (2008) 4458
12. Ontanon, S., Lee, Y.C., Snodgrass, S., Bonfiglio, D., Winston, F., McDonald, C., Gonzalez, A.: Case-based prediction of teen driver behavior and skill. In: Case-Based Reasoning Research and Development, ICCBR 2014, Berlin, Springer (2014) 375389
13. Cordier, A., Lefevre, M., Champin, P.A., Georgeon, O., Mille, A.: Trace-based reasoning modeling interaction traces for reasoning on experiences. In: Proceedings of the 2014 Florida AI Research Symposium, AAAI Press (2014) 363368
14. S. Windmann, A. Maier, O. Niggemann, C. Frey, A. Bernardi, Ying Gu, H. Pfrommer, T. Steckel, M. Kruger and R. Kraus, "Big Data Analysis of Manufacturing Processes," in 12th European Workshop on Advanced Control and Diagnosis, 2015.
15. Mller, G., Bergmann, R.: Workflow Streams: A Means for Compositional Adaptation in Process-Oriented CBR. In: Proceedings of ICCBR 2014. Cork, Ireland, 2014

A Collaborative CBR Recommender System to Support Patients, its Relatives and Caregivers in Chronic and Palliative Care

Sara Nasiri

Department of Electrical Engineering & Computer Science, University of Siegen
Institute of Knowledge Based Systems & Knowledge Management
Hölderlinstr. 3
57076 Siegen
sara.nasiri@uni-siegen.de

1. Introduction

In the last decades, many medical assistance systems have been developed, and the interest in computer-aided problem-solving in medical and healthcare is constantly growing. Most of the software systems in this domain focus on decision support and recommendation of effective medication for patients. The combination of statistical analysis and case-based reasoning can facilitate a better medical diagnosis [1] [2] [3]. Within the case comparison mechanism of CBR, feature selection, similarity measurement, and adaptation methods play an important role to retrieve and revise cases. In this research, DePicT (**De**tect and **P**redict diseases using **i**mage **c**lassification and **T**ext Information from patient health records) uses image interpretation and word associations for feature selection and recommendation of medical solutions [4]. All gathered patient records are stored in relational databases as structured or closed-format (e.g. parameters and statistics), or unstructured or open-format e.g. texts and images. For example, images of affected areas of a melanoma skin cancer can contribute and support early stage diagnosis. Also, further information on answering questions or writing a statement about the patient's health condition is added to the knowledge base. Domain Experts can validate and verify the collected information and also update the case-base to correct the data records of patients. In the other hand, more over than assisting for detecting and predict the disease, the Vocational Educational Training (VET) and Technology Enhanced Learning (TEL) [5] is a research field which is investigated continuously. DePicT CLASS (**De**tect and **P**redict diseases using **i**mage **c**lassification and **T**ext information in **C**ase-based **L**earning **As**sistant **S**ystem) is a CBR system by enrichment of cases with learning materials (e.g. reference images and textbook) [6]. It is utilized smart (knowledge-based) and accessible systems to provide vocational educational learning opportunities and achieving higher education. CBR is applied in various problem-solving domains, and it is appropriate in medicine to integrate the system and for explicit experience, cognitive adequateness, the duality of objective/subjective knowledge, and to extract subjective knowledge [7]. Design and development of the DePicT and DePicT CLASS are the main contributions of this investigation. It is a case-based system which uses DePicT Profile Matrix of the association strength between title phrase and identified keywords of cases. Making experiments to validate the research and this recommender system lead us to do it in the skin and brain diseases as patient assistance system-DePicT CBMelanom and caregiver learning assistant system-DePicT Dementia CLASS, respectively.

2. Research Questions and Aims

This section presents the research questions which have been reached by my doctoral thesis:

1. How can we extract knowledge from healthcare processes and stakeholders to find the gap in the current system to create the desired system with considering requested a change?
2. How is it possible to establish an assistant system and utilizing data from the network communication between patients, caregivers and doctors contributing to a better understanding of their challenges?
3. How can we improve user experience with using Case-based systems?
4. How to evaluate the adaptation mechanism?
    a. A new proposal that word association profile can follow adaptation on the most similar cases. The practicalities of this will be discussed.
    b. A system is learned which guides manual adaptation of similar references retrieved from the case base. (e.g. learner can rank the best references)
    c. In the context of adaptation, it is compared with the:
        i. Information extraction using domain expert & stakeholders
        ii. Information extraction using references.
        iii. Statistical approaches (e.g. value),
        iv. Statistical approaches enhanced by learned knowledge. (e.g. grade of ranking)

The approaches to addressing the research questions and the explanation of how the remaining aspects of these research questions are going to be investigated are described in the next sections.

3. Proposed Plan of Research

This research divided to a preliminary concept which is DePicT and the educational concept which is DePicT CLASS recommender procedure. In these concepts, we focus on the combination of the textual and structural case based reasoning and utilizes word association method which is called CIMAWA [8] to find the word association strength between case title and case features which identified keywords.
DePicT [4] is a conception of a knowledge-based system for the identification and diagnosis of diseases. It utilizes the graphical and textual similarity measurements of non-image and image information which are Tverskey's similarity measure [9], Frucci et al.'s image dissimilarity measure [10]. So, DePicT CLASS [6] is able to search for references based on the comparison of word association profiles of identified keywords to find the best similar ones with the request. DePicT CLASS is a Case-based Learning Assistant System which is the application of the DePicT concept. To answer the research questions which are explained in the previous section, two use cases are investigated. The prototype of DePicT CBMelanom is developed which is briefly explained in the next section and DePicT Dementia CLASS is fully implemented based on the retrieval and adaptation mechanism of DePicT CLASS [6] and evaluated based on the test problems from

Alzheimer and dementia forums and homepages [11]. This research proposes a new adaptation method for dementia vocational educational training which uses the WHO Framework for the International Classification of Functioning, Disability, and Health (ICF) [12] word association strength of the DePicT Profile Matrix [6], [11]. It is utilized abstraction adaptation to characterize each case by an identified keywords list which is associated with each dementia disease and compositional adaptation to computing a value for each reference from the most similar cases. Based on the definition of Kolodner [13], DePicT CLASS adaptation is also categorized as a particular type, while it is considered the collaborative recommendation of users by ranking the references, adding the images tags, suggesting images impact factors, sending the feedback to contribute to the reference collection. Therefore, the DePicT CLASS adaptation mechanism has a combination of value comparison based on the requested word association profiles and manual adaptation based on user collaborative recommendation e.g. learner can rank the best references and learning materials based on their understanding and requirements. In future, the other parameters of caregiving e.g. their challenges and task's difficulties will be considered. Moreover, for completing the evaluation phase, it will be tested with a domain of informal caregivers. For further development of an application of the DePicT concept, DePicT CBMealnom will be developed based on the 7-point checklist (7PCL) which has been recommended by NICE (2005) and the case base is modifying based on the Melanoma Datasets. The future work is addressing the fourth research question in this use case.

4. Description of Progress to Date

DePicT CBMelanom is utilized by patients who has skin problems. While a user can not necessarily be able to formulate the question in a machine-readable form which is ready for the CBR system, Conversational Case-Based Reasoning provides a question dialog to guide users to describe their problem incrementally through an answering procedure [14, 15]. Its prototype utilizes myCBR tool [16] to create the CBR system for early detection of skin cancer. Based on the report of American cancer society's cancer facts and figures 2016 [10], "Melanoma accounts for only 1% of all skin cancer cases, but the vast majority of skin cancer deaths. In 2016, an estimated 10,130 deaths from melanoma and 3,520 deaths from other types of skin cancer (not including KC) will occur." Therefore, early detection is crucial in this kind of cancers and "the best way to detect it early is to recognize new or changing skin growths, particularly those that look different from other moles [10]." Even after treatment, it is imperative that patients keep their medical history and records [11]. Therefore, DePicT Case-based Melanom (DePicT CBMelanom) illustrates how Melanoma is detected and predicted utilizing conversational case-based reasoning.

DePicT CLASS of Dementia is used and updated by caregivers and domain experts. It enables caregivers and patients' relatives to find their learning materials and references which address the problems that they are looking for. Although the increasing prevalence of dementia poses a major challenge for global health at multiple levels [17], CBR is applied in the care of Alzheimer's disease patients from 2001 [18]. "Dementia encompasses a range of neurological disorders characterized by memory loss and cognitive impairment. In 2015, almost 47 million people worldwide were estimated to be affected by dementia, and the numbers are expected to reach 75 million by 2030, and 131 million by 2050, with the greatest increase expected in low-income and middle-income countries [19]. Since the development of ICF [12], several projects for specific health conditions and disabilities are defined to develop core sets of ICF codes. DePicT CLASS uses DePicT Profile Matrix weights of the association strength between title phrase and identified keywords of cases (including references) which are dementia related diseases and ICF parameters, respectively. In this analysis, the references and learning materials with high valued keywords in word association profiles from the most similar cases are recommended to the selected case. This research proposes a new abstraction, compositional and collaborative adaptation method for medical vocational educational training which uses the calculated word association strength of the DePicT Profile Matrix [6]. DePicT Dementia CLASS is used and updated by caregivers and domain experts. It enables caregivers and patient's relatives to find their learning materials and references which address the problems that they are looking for. Case formation identifies the requested keywords and assigning their values based on DePicT Profile Matrix. These 111 ICF parameters are searched in 40 dementia and caregiving books and handbooks to create the large document as a reference of DePicT Profile Matrix. DePicT Dementia CLASS experimentally evaluated adapted references compared to the retrieval only references using its similarity measurement [6]. We have used two rates to investigate the hypothesis which is DePicT Dementia CLASS is able to select cases (three most similar ones) which can be adapted more related to the user request in comparison with the retrieval only references. The attract rate is defined based on the ratio of the value of references to their rank. In this way, DePicT CLASS compares reference ranking which is enhanced by learned knowledge of users. Besides the attract rate, for evaluating the adaptation results, the adaptation rate (adapt rate) is defined based on the ratio of retrieval only references to the total number of associated references. Thus, the recommendation of selected case is arranged based on the combination of high-value and ranked references.

References

[1] N. Choudhury and S. A. Begum, "A Survey on Case-based Reasoning in Medicine," *(IJACSA) International Journal of Advanced Computer Science and Applications,* vol. 7, no. 8, pp. 136-144, 2016.

[2] R. Schmidt, S. Montani, R. Bellazzi, L. Portinale and L. Gierl, "Casedbased reasoning for medical knowledge-based systems," *International Journal of Medical Informatics,* vol. 64, no. 2, pp. 355-367, 2001.

[3] M. U. Ahmed, S. Begum, E. Olsson, N. Xiong and P. Funk, "Case-Based Reasoning for Medical and Industrial Decision Support Systems," in *Successful Case-Based Reasoning Application*, S. Montani & L.C. Jain (Eds.), SCI 305, Springer-Verlag Berlin Heidelberg, pp. 7-52, 2010.

[4] S. Nasiri, J. Zenkert and M. Fathi, "A medical case-based reasoning approach using image classification and text information for recommendation," in *Advances in Computational Intelligence*, Springer, pp. 43-55, 2015.

[5] M. Kalz and M. Specht, "Assessing the cross disciplinarity of technology enhanced learning with science overlay maps and diversity measures,," *British Journal of Educational Technology* , vol. 45, no. 3, pp. 415-427, 2014.

[6] S. Nasiri, J. Zenckert and M. Fathi, “Improving CBR Adaptation for Recommendation of Associated References in a Knowledge-based Learning Assistant System,” *Neurocomputing*, vol. 250, DOI: http://dx.doi.org/10.1016/j.neucom.2016.10.078, pp. 5-17, 2017.

[7] M.-J. Huang, M.-Y. Chen and S.-C. Lee, “Integrating data mining with case-based reasoning for chronic diseases prognosis and diagnosis,” *Expert Systems with Applications* , vol. 32, no. 3, pp. 856-867, 2007.

[8] P. Uhr, A. Klahold and M. Fathi, “Imitation of the human ability of word association,” *International Journal of Soft Computing and Software Engineering(JSCSE)* , vol. 3, no. 3, pp. 248-254, 2013.

[9] A. Tversky, “Features of similarity,” *Psychological review 84,* vol. (4), pp. 327-350, 1977.

[10] M. Frucci, P. Perner and G. Sanniti di Baja, “Case-Based Reasoning for Image Segmentation Segmentation by Watershed Transformation,” in *Case-Based Reasoning on Images and Signals*, Springer-Verlag Berlin Heidelberg 73, pp. 319-353, 2008.

[11] S. Nasiri and M. Fathi, Case Representation and Similarity Assessment in a Recommender System to Support Dementia Caregivers in Geriatric and Palliative Care, Submitted to ICCBR, 2017.

[12] WHO, “International Classification of Functioning, Disability and Health (ICF),” 2001.

[13] J. Kolodner, Case-based Reasoning, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[14] M. Gu and A. Aamodt, “Dialog Learning in Conversational CBR,” American Association for Artificial Intelli-gence (www.aaai.org), 2006.

[15] M. M. Richter and R. O. Weber, Case-Based Reasoning a Textbook, Springer-Verlag Berlin Heidelberg , 2013.

[16] myCBR, “DFKI (Germany) - UWL (UK),” 2015. Available: http://www.mycbr-project.net.

[17] M. Ienca et. al., "Intelligent Assistive Technology for Alzheimer’s Disease and Other Dementias: A Systematic Review," *Journal of Alzheimer’s Disease,* Vols. 56, no. 4, DOI: 10.3233/JAD-161037, pp. 1301–1340, 2017.

[18] S. Marling and P. Whitehouse, “Case-Based Reasoning in the Care of Alzheimer's Disease Patients,” *In Lecture Notes in Computer Science*, 2001.

[19] B. Winblad et. al., "Defeating Alzheimer’s disease and other dementias: a priority for European science and society," *Lancet Neurology Commission,* vol. 15, pp. 455–532, 2016.

Case-based Interpretation of Best Medical Coding Practices — Application to Data Collection for Cancer Registries

Michael Schnell[1,2]

[1] Epidemiology and Public Health Research Unit, Department of Population Health, Luxembourg Institute of Health, 1A-B, rue Thomas Edison, L-1445 Strassen, Luxembourg, michael.schnell@lih.lu

[2] UL, CNRS, Inria, Loria, F-54000 Nancy, michael.schnell@loria.fr

1 Introduction

Currently, cancer remains one of the main causes of death. To assist in the fight against this disease, cancer registries are being used. A cancer registry is a systematic, continuous, exhaustive and non redundant collection of data about cancers diagnosed and/or treated in a country or region. The collected data is defined in international standards with common terminologies (e.g., the International Classification of Diseases, commonly abbreviated as ICD [7]) and associated to best medical coding practices. Unfortunately, these practices and standards are very complex, making it difficult for operators, i.e. medical staff in charge of collecting and coding the data, and coding experts to apply them efficiently and consistently.

The aim of this research is to tackle this complexity, by assisting both operators and coding experts in the interpretation of best medical coding practices.

For the Luxembourg National Cancer Registry (NCR), a ticketing system has been implemented for operators. When they encounter a difficult coding problem, they can ask questions through this system and coding experts provide individual answers. Interesting questions are later presented and discussed in regular training sessions for operators. Coding experts rely on their medical knowledge and their understanding of the coding standards and best medical coding practices to answer the questions. However, it is crucial for cancer registries to have a consistent coding of the data, meaning that two similar patients should be coded similarly. Thus, two similar coding questions should have similar answers. For that reason, the coding experts must also take into account previous questions to answer new questions.

Section 2 presents the research plan, followed by a review of the current progress in section 3. Finally, section 4 outlines the remaining work.

2 Research plan

The research plan contains the following main work packages: an analysis of the questions answering approach and design of a coding solving method and a question management tool, an implementation of the tool and an evaluation of the impact of the proposed tool.

The analysis and design package contains the following areas of work:

- a literature review for case-based reasoning and the combination with other methods, e.g., rule-based reasoning, belief merging or formal argumentation;
- a state of the art for coding assistants and decision support systems;
- a analysis of the questions asked by operators of the NCR and the answers provided by experts, as well as their solving process;
- functional design of a question management tool and technical choices for the implementation.

With the results of the analysis, a implementation of the designed software will be provided, starting with a prototype for a limited test usage before transitioning to a more general usage.

Lastly, an evaluation of the impact of the implemented assistant, notably to assess the evolution of the expert and operator workloads, is planned.

3 Current progress

The first step of this research is to analyze the coding process and the question solving process. Therefore, a dozen tricky questions and several easier questions have been discussed with the coding experts of the NCR. Given the similarities in the solving process of the experts and case-based reasoning, the proposed approach is adapted from the 4-R cycle presented in [1] and the knowledge containers presented in [6]. Other approaches have been considered (e.g., automatic coding [5]), but are not being pursued at the moment.

The approach proposed by this research uses arguments for the solving process. Indeed, when answering a case, experts often point out which arguments (pros and/or cons) have been identified and which solution they support. This is very helpful for operators, as it gives them insights into the reasoning process and allows them to learn more quickly. In order to incorporate this user guidance and solution explanation into our approach, the retrieval step uses the arguments of a source case to find the best match for the given target problem. The approach has been described in more detail in a paper submitted to ICCBR 2017. A prototype developed for the paper (shown in figures 1 and 2) focuses on structured questions asked by operators and the solutions provided by the system.

Arguments have already been used in CBR, but not to identify similar cases. For example, in [3] and [4] arguments are generated and used to explain the inferred solution of the target problem.

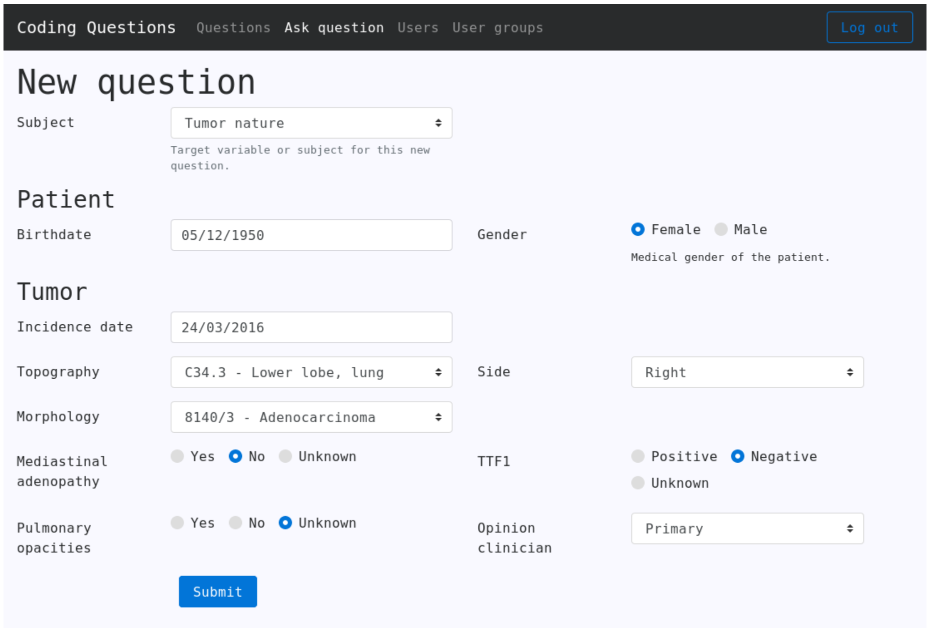


Fig. 1. Form used for the question asked by operators and to describe the target problem.

To facilitate the handling of the operators' questions, it was decided to structure the questions rather than apply natural language processing methods. Figure 1 shows an example of a structured question. Still, there are plenty of items that can be relevant for the various questions, making it very difficult to define every possible data item. Thus, only the most important subject are completely structured, i.e. the data asked of the operator is almost exhaustively defined. The assistant proposed by this research project will only handle structured questions. The remaining unstructured questions, i.e. questions where the operators describes his problem using free text, will continue to be answered by the coding experts, with little change to the current situation.

4 Future work

Once the prototype has been tested, a first version of the ticketing system for the coding questions will be developed. By the end of the year, this first version should be tested by the operators of the NCR. This version will be evaluated, notably to determine the impact of the system. Several types of criteria can be measured (e.g., quality of the coded data, workload of the coding experts). The final list of criteria will be determined during the implementation of the ticketing system. Alongside, other avenues are considered:

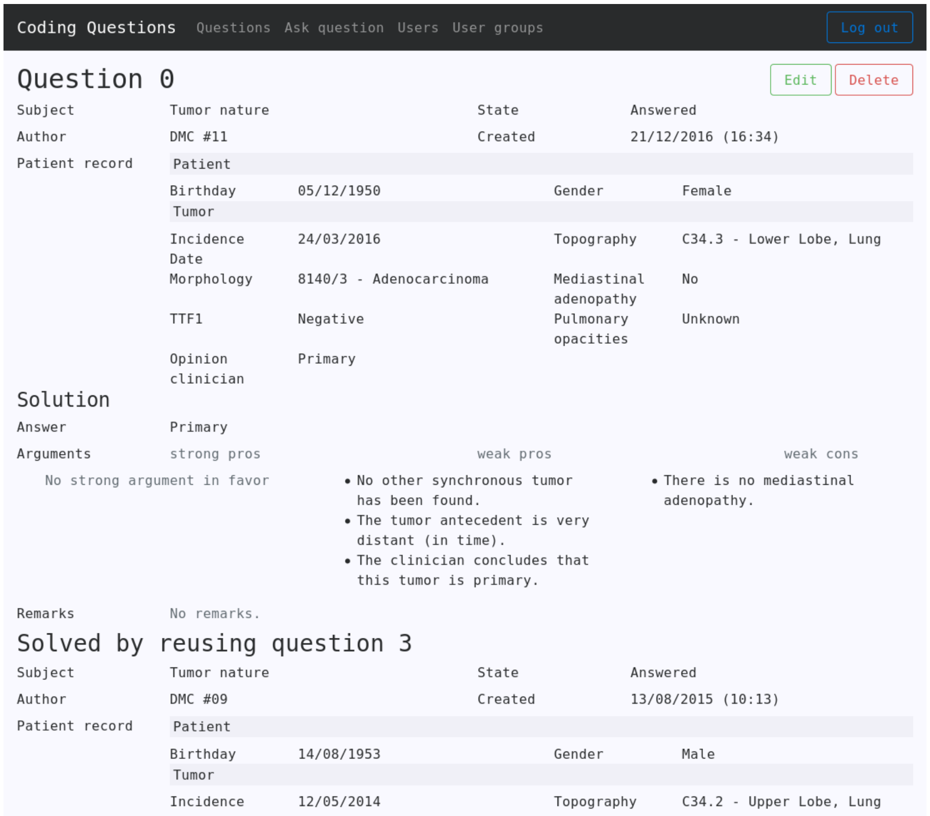


Fig. 2. Summary of the described target problem and the proposed solution. The retrieved source case is shown similarly to the target problem.

– increasing the efficiency of the solving algorithm (e.g., combining arguments using logic operators like *and*, reusing arguments from several source cases);
– adding a conversational approach where the system can ask for additional information about a given patient (to reduce the amount of information asked to only relevant data);

The following problems, though interesting, will probably not be researched in depth during this project:

– adding a confidence indication of the given result to help operators and coding experts distinguish between tentative solutions and validated solutions;
– taking into account the evolution of the coding standards [2].

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches (1994)

2. Cardoso, S.D., Pruski, C., Da Silveira, M., Lin, Y.C., Groß, A., Rahm, E., Reynaud-Delaître, C.: Leveraging the Impact of Ontology Evolution on Semantic Annotations, pp. 68–82. Springer International Publishing, Cham (2016)
3. McSherry, D.: Explaining the Pros and Cons of Conclusions in CBR. In: European Conference on Case-Based Reasoning. pp. 317–330. Springer (2004)
4. Murdock, J.W., Aha, D.W., Breslow, L.A.: Assessing elaborated hypotheses: An interpretive case-based reasoning approach. In: International Conference on Case-Based Reasoning. pp. 332–346. Springer (2003)
5. Pestian, J.P., Brew, C., Matykiewicz, P., Hovermale, D.J., Johnson, N., Cohen, K.B., Duch, W.: A shared task involving multi-label classification of clinical free text. In: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing. pp. 97–104. Association for Computational Linguistics, 1572411 (2007)
6. Richter, M.M., Weber, R.O.: Case-based reasoning: a textbook. Springer Science & Business Media (2013)
7. World Health Organisation: The ICD-10 classification of mental and behavioural disorders. Geneva: World Health Organisation (1992), `http://www.who.int/classifications/icd/en`

Research Summary: Knowledge Transfer in Artificial Learning

Pierre-Alexandre Murena

Télécom ParisTech - Université Paris Saclay,
46 rue Barrault, 75013 Paris, France,
murena@telecom-paristech.fr

1 Introduction

This document presents the main research problems addressed during my PhD studies. All these researches are led inside the two teams DBWeb in Télécom ParisTech and LInK (Learning and Integration of Knowledge) in AgroParisTech, both located in Paris, and supervised by Pr. Jean-Louis Dessalles and Pr. Antoine Cornuéjols.

My researches focus on learning theory both in the perspective of symbolic machine learning and of learning in continuous domains. I aim at finding an information-theoretic principle guiding information transfer in learning.

The start point of these researches is the idea that most machine learning takes a strong stationary hypothesis for granted. The general framework of statistical learning (mainly supervised and semi-supervised learning) considers two data sets: a *learning data set* (from which the concepts have to be learned) and a *test data set* (on which quality of the learned concepts is evaluated). The key idea of current learning methods and theories is to assume that *training data and test data are independent and identically distributed (i.i.d.)*. However this strong hypothesis does not hold in many cases: either the data generation process evolves over time (aging effect, trending effect...) or the data belong to a different domain. Because similar questions of transfer and domain adaptation had already been addressed in analogical reasoning, we proposed to use an approach based on Kolmogorov complexity instead of probabilities. Kolmogorov complexity is a measure of the information contained inside an object. The use of Kolmogorov complexity in machine learning is accepted by the community, but mainly in a stationary point of view (when the key concept does not vary); we proposed to extend its use to non-stationary environments, in the same way as done in analogical reasoning. A presentation of these issues is given in section 2.

The strong similarity between transfer learning and analogical reasoning led me to consider this issue in my researches. Analogical reasoning consists in situations of the form "*'b' is to 'a' as 'd' is to 'c'*". Because its value has already been demonstrated, I focus on Hofstadter's micro-world, made up of strings of alphabetical characters that can be described with simple concepts like 'predecessor', 'successor' or 'repetition'. The use of Kolmogorov complexity for analogical reasoning had already been considered, but our approach is slightly different. We developed a small descriptive language for Hofstadter's problems and convert it into a binary code, the length of which corresponds to Kolmogorov complexity. Our work on analogy is presented in section 3. Finally, because of its very global perspective, our research topic leads naturally to collaborations on various topics related to learning. This side aspect of my research is presented in section 4.

2 A global approach of learning

Statistical machine learning in its current form is often considered to be based mainly on three inductive principles: Empirical Risk Minimization (ERM), Bayesianism and Minimum Description Length (MDL). The validity of ERM has been demonstrated under the strong i.i.d. assumption using several frameworks, all inspired by the Probably Approximately Correct learning framework. Besides strong links have been stated between Bayesianism and MDL.

When the i.i.d. hypothesis is not verified, these three principles are not valid anymore and have to be replaced by new principles. Exploring this direction, we considered the most straightforward transfer learning problem and the classification problem, the purpose of which is to associate data to labels. Given source data X_S associated to their classes Y_S and a target data X_T, we aim at finding the corresponding classes Y_T. The idea is to find a classification function β_S such that $\beta_S(X_S) = Y_S$ and to transfer this function into a classification function β_T available on the target data X_T. Because this problem is the same as analogical reasoning, we used a simplification of the general MDL principle in the context of analogy [2].

The mathematical tool used to measure the *description length* in MDL is Kolmogorov complexity [4]. Kolmogorov complexity (also called algorithmic complexity) of an object x is an information theoretic measure defined as the minimal length of a program defined on a Universal Turing Machine (UTM) and the output of which is object x. This quantity can be shown to be machine independent, but non calculable.

The idea we developed is to consider an upper-bound of this complexity based on a restricted Turing machine. The choice of a restricted Turing machine corresponds to an inductive bias, inevitable in any inductive reasoning (see for example the *no-free-lunch theorem*). This choice also raises the problem of machine dependency which seems crucial in human learning.

Our first contribution is a direct use of analogical MDL in the context of transfer learning [5]: I presented a two-part MDL equation based on a data representation called *model*. A model is any object which may be used to compress data. In transfer learning, our purpose is to infer a source model M_S and a target model M_T:

$$\min_{M_S, M_T} \quad C(M_S) + C(X_S|M_S) + C(Y_S|M_S, X_S) + C(M_T|M_S) + C(X_T|M_T) \quad (1)$$

where $C(.)$ designates Kolmogorov complexity, X the data, Y_S the source labels and M the model. This equation applies both for continuous data (X is a matrix, the rows of which correspond to a vector data point) and for symbolic data (X is a sequence of symbols, for example a sequence of letters). The different terms in the equation present a strong similarity with usual machine learning terms: $C(M)$ corresponds to a model penalization based on its complexity; $C(X|M)$ corresponds to a likelihood term, ie. a fitness of the model toward data; and the term $C(Y|M,X)$ corresponds to an empirical risk. In the paper, we also proposed experimental validations on two toy data sets with a simple prototype-based model. They present good results and high performance on these data sets.

A direct variant of this formula has been proposed for incremental learning [6]. In incremental learning, the system faces a sequence of questions $X_1, X_2, \ldots$ and has to find the solution to each problem one by one. The model used to describe data can be updated if it is outdated and does not correspond to the current data anymore. We propose the following simplified MDL objective:

$$\min_{M} \quad \sum_{t} C(M_t|M_{\Delta_t^{-1}(\{1\})}) + C(X_t|M_t) + C(\beta_t|M_t, X_t) + C(Y_t|M_t, X_t, \beta_t) \quad (2)$$

where Δ_t is a model association function such that $\Delta_t(u) = 1$ if model M_t can be described with model M_u, and $\Delta_t(u) = 0$ otherwise. The consistency of our framework with existing heuristics state-of-the-art methods has been established, as well as the validity of a naive algorithm based on the same neural model as presented for transfer learning.

The successful results obtained with MDL so far encourage future research tracks. Several problems emerge from the developed framework. First, the transfer objective 1 is valid for one target only. In practice, several target problems may occur, hence a multi-target variant of transfer has to be given. In particular, i.i.d. hypothesis consists in assuming infinitely-many targets with specific distributions. I am currently exploring an approach based on Pareto-optimality, implying a prior over the future and thus a new learning concept: *concern for future question.*

Another question of interest is the theoretical validity of such approaches. Unlike statistical learning which has a clear measure of quality (given by the risk), an approach based on MDL does not present any natural quality measure. Such a function has to be found before an equivalent of PAC learning can be developed. Additionally, incremental learning methods do not have access to the whole objective function 2 at all steps: only local optimizations are possible. A measure of the impact of this algorithmic simplification appears as a direct consequence. Finally, we aim at finding a geometric interpretation of these equations. An interesting track is offered by the domain of information geometry and probability distribution manifolds. Under some specific conditions, Kolmogorov complexity may be associated to a probability distributed, hence considered in the perspective of information geometry.

3 Approaches of analogy

Because of its crucial role in my researches on learning, I attach great importance to studying analogical reasoning. For now on, my researches focused on Hofstadter's micro-world [3] which presents highly general characteristics of analogical thought. I will work on this research track with Dr. Laurent Orseau.

Preliminary works have already given insightful results and promising perspectives. I chose to work on the development of a small prototype language generating Hofstadter's analogies (of the form **ABC : ABD :: IJK : IJL**, which has the be read "**ABD** is to **ABC** what **IJL** is to **IJK**"). Among other specifications, the proposed language had to be generative, ie. describe a dynamic generation rather than a static description (as opposed to the description in [2] for instance). For example, the string **ABC** will be generated by the program `alphabet,sequence,3`, which can be read as "*Consider the alphabet and take the sequence of first three letters*". Once such a language is defined, it is turned into a prefix-delimited binary code, the length of which measures an upper-bound of Kolmogorov complexity.

A more elaborated and general version of the language has been recently proposed. This memory-based language offers a flexibility in the management of prior knowledge of the user and offers a simple way to set priority to operations: its grammar enables any possible operator, as long as the operator can be put in long-term memory. The complexity of an element in memory is defined as the complexity of its depth in memory. A more rigorous presentation of this new framework including the considerations on memory will be presented at ICCBR 2017 Computational Analogy Workshop [7].

We propose a validation of our approach with a human experiment. In an online survey, we submitted a few Hofstadter's problems and asked participants for their most intuitive solution. We shew that the majority answers correspond to local minima of Kolmogorov complexity. These results are not yet published.

The proposed approach offers a tool to compare two results of an analogical problem when the generative instructions are given to the system. A first logical direction is to provide automatic instruction generation, hence software able to produce an optimal generative instruction for any complete analogy. Once this will done, I have to find a solution to an analogical problem (eg. find the best solution to **ABC:ABD::IJK:?**): because the space of solutions is infinite, it cannot be explored naively and research biases have to be found. In order to address these issues, I am currently working on a Python interpreter for the developed language.

4 Collaborations and side problems

In the context of my research, I have the opportunity to collaborate on several projects related to non-stationarity and transfer.

A first project is led jointly with Dr. Jérémie Sublime (ISEP) and Dr. Basarab Matei (Université Paris 13) and concerns collaborative clustering. Classic clustering consists in associating similar data together in *clusters*. Collaborative and multi-view clustering is a framework in which several clustering algorithms are involved and try to influence each other. The algorithms do not produce the same number of clusters, the same underlying model nor the same final solution. This problem is closely related to transfer because it involves sharing information between several different domains. A first contribution has been proposed using operational research tools to select relevant collaborators in an existing probabilistic framework [9]. A brand new approach, based on complexity, is being developed: we expressed the problem of collaborative clustering in terms of data compression and worked on minimal assumptions to obtain a tractable model. This new perspective offers a theoretical background for a wide range of heuristic state-of-the-art approaches and inspires a new algorithm [8].

A new related collaboration has been engaged recently with Dr. Cristina Manfredotti, Pr. Juliette Dibie (AgroParisTech) and Dr. Fatiha Sais (LRI), exploring common approaches in transfer learning and structural mappings in knowledge bases (ontology alignment).

Finally, I am exploring the problem of Transfer Learning using boosting algorithms with Pr. Antoine Cornuéjols and Sema Akkoyunlu. The use of boosting may offer new perspectives on transfer in general and be beneficial to my understanding of transfer mechanisms. A first contribution has been submitted [1].

References

1. Cornuéjols, A., Akkoyunlu, S., Murena, P.A., Olivier, R.: Transfer learning by boosting projections from the target domain to the source domain, submitted to NIPS 2017
2. Cornuéjols, A., Ales-Bianchetti, J.: Analogy and induction : which (missing) link? In: Workshop “Advances in Analogy Research : Integration of Theory and Data from Cognitive, Computational and Neural Sciences”. Sofia, Bulgaria (1998)
3. Hofstadter, D.: The copycat project: An experiment in nondeterminism and creative analogies. AI Memo 755, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (1984)
4. Li, M., Vitanyi, P.M.: An Introduction to Kolmogorov Complexity and Its Applications. Springer Publishing Company, Incorporated, 3 edn. (2008)
5. Murena, P., Cornuéjols, A.: Minimum description length principle applied to structure adaptation for classification under concept drift. In: 2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016. pp. 2842–2849 (2016)
6. Murena, P.A., Cornuéjols, A., Dessalles, J.L.: Incremental learning with the minimum description length principle, accepted in International Joint Conference on Neural Networks 2017
7. Murena, P.A., Dessalles, J.L., Cornuéjols, A.: A complexity based approach for solving hofstadter’s analogies, accepted at Computational Analogy Workshop, ICCBR 2017
8. Murena, P.A., Sublime, J., Matei, B., Cornuéjols, A.: Multi-view clustering with the principle of minimum description length, submitted to ECML 2017
9. Sublime, J., Matei, B., Murena, P.A.: Analysis of the influence of diversity in collaborative and multi-view clustering, accepted in International Joint Conference on Neural Networks 2017

PhD project plan - Exploiting data to support operations of EXPOSED aquaculture installations

Bjørn Magnus Mathisen

No Institute Given

1 Introduction

Aquaculture is an increasingly important industrial sector in Norway. The Norwegian Ministry of Fishery and Aquaculture states that its a long term goal for aquaculture is a five fold increase in production by 2050 [1]. This stands in direct contrast with recent Norwegian news[2] which states that aquaculture in Norway is facing a growing challenge with regards to fish disease and environmental impact.

To help the industry reach the goal set by the Ministry of Fishery and Aquaculture of a five fold increase - without increasing the environmental impact - we need to increase the amount of locations suitable for modern aquaculture development. The SFI project EXPOSED[3] is trying to do this by researching and developing technologies that enable aquaculture to operate at more exposed (offshore and subject to harsh weather) locations.

A part of the technology needed for this is increased level of monitoring and decision support. Making use of the increasing amount of data gathered on these aquaculture installations to automate and support personnel would decrease cost and the amount of hours spent on the installations thus decreasing the risk for the personnel at these more exposed locations. Decision support systems within aquaculture traditionally employ numerical models (e.g. [7]). With the increasing amount of data being gathered at these installations, using machine learning to create models from the data would be a great addition to these models as a part of a decision support system.

Machine learning is a promising field of research. Most recently shown by the rising field of deep learning. Deep Learning has been employed for data analysis within different domains, including, but not limited to; speech recognition [5, 10], object recognition [4] and text processing [2]. Recently it has also been tried as methods for improving computer-chess[4] and computer-go [12]

Given enough data and targeted at the correct problems (e.g. where learning patterns across huge amounts of data will lead to a solution) these methods excel. However if the data is sparse these methods can be hard to utilize. Another missing feature from these types of methods is explainability.

Methods like deep learning (and other sub-symbolic methods) are not easily explained. It can be done through analyzing activation patterns in the neural networks etc, but it is undecipherable to anyone but experts in the field.

Explainability is a key factor in making the human operator trust the decision/operator support system and other machine learning methods are easier to explain and understand, such as Bayesian networks[15] or to a lesser degree Evolutionary Algorithms[11, 6].

Case-based reasoning (CBR)[1] is an example of another method that is based on symbols and knowledge (also known as a knowledge-based reasoning method). As the method is heavily inspired by cognitive models, the method is more easily explainable. CBR in combination with knowledge engineering in cooperation with human experts can even achieve high performance without the

[1] http://www.vg.no/nyheter/innenriks/naeringsliv/slik-kan-fremtidens-lakseoppdrett-se-ut/a/23568283/

[2] http://www.vg.no/nyheter/innenriks/her-er-kartet-som-faar-ekspertene-til-aa-slaa-laksealarm/a/23598279/

[3] http://exposedaquaculture.no/

[4] http://erikbern.com/?p=841

presence of large data-sets. The method has been tested with success in several domains such as oil & gas[8] and fish farms [16]. This method is typically less dependent on a large data corpus than other methods such as neural networks. This property enables us to use CBR as a method for predicting rare events.

To reach the goals set for the SFI project: improved quality, safety and efficiency in exposed fish farming operations - the solution should utilize the strength of all of these methods. E.g. deep neural networks may be used to detect anomalies which correlates or causes a change in site state which may be extracted and refined as a case and stored in a case-base. This could then be used by a CBR system to predict future anomalies with a greater ability to explain how it came to this conclusion.

Machine learning has already been applied to fisheries and aquaculture through decision support, examples of this is; Operational support in fish farming through using CBR[16], decision support model for fisheries management in Hawaii[13], decision support system for fish disease/health management[17], decision support for sustainability[9]. Our work will build on these results, applying what has been learned in previous studies. Decision support systems within aquaculture has been used but a predominant part of the decision support systems described in literature[3, 7] only employ user input and numerical models. Our system will employ data-based machine learning methods that build models and tries to predict future states, knowledge based methods that improves explainability and can support prediction of rare events, all in addition to the traditional numerical models.

The goals of this PhD project is to contribute to a subset of the main goals set for the SFI-center as mentioned above -

Goal ($G1$) : *This PhD project will study AI methods aiming for safe and sustainable fish production at exposed aquaculture sites through utilization of sensor data as well as human experiences to develop and test a system for monitoring, prediction, and operational decision support.)*

2 Objectives

The work done in this project will contribute to machine learning and artificial intelligence via testing the applicability and performance of AI/ML methods in the aquaculture domain. On the other side, the project will advance decision support systems within the aquaculture domain via testing how ML methods can increase their performance and usefulness. However many of today's popular machine learning methods requires large databases of instances to be trained properly, but many of the events that operators wants to avoid on a installation are rare and as such the data about them is sparse. More specifically the measurable academic objectives of this PhD project is:

1. Establish the current state of the art in terms of decision support systems within aquaculture and the role of machine learning in these decision support systems.
2. Create and test machine learning methods that uses data from aquaculture installations and contributes to a decision support system for such a site. Typically these methods will try predict future states of a aquaculture installation and this prediction can be used by a decision support system.
3. Create and test knowledge-based methods such as CBR for detecting and predicting rare events in the aquaculture domain.

3 State of the PhD project

The EXPOSED SFI project is currently gathering sensor data and operational data from exposed sites. Table 1 lists some of the data gathered from the pilot sites in the EXPOSED project.

Environmental measurements	Installation	Production data
Wind (max and avg)	Movement (accelleration)	Fish mortality
Waves (max and avg)	Anchor load	Fish grouping
Current (direction and speed at 3 depths)	Operational status[5]	Feeding
Temperature	-	-

Table 1: Table listing some of the data parameters gathered from sites included in the SFI EXPOSED project.

Not all of this data is gathered for every aquaculture site included in the project. As a result some sites will have more data parameters gathered than others. In addition the instrumentation of these sites are a central part of the project as such, and is thus evolving, increasing the amount data gathered at each site. At this point in the project we are focusing on implementing a prototype framework for automatically importing these data streams. As a part of this framework we implemented a way to easily test different machine learning methods w.r.t. using the different data parameters to predict future states of a aquaculture site. We have performed two tests in this regard which we present in the following subsection.

3.1 Predicting installation movement

In this experiment we tried to predict the movement of an aquaculture cage structure based on wind data gathered from a buoy that is situated close (within 100m) of the cage. To approximate the level of movement on the cage structure we calculated the variation of the x axis of the accelerometer mounted on the cage structure over the span of an hour (36000 data points, 10 hz). The prediction was made with a neural network with inputs being Wind Direction, Wind Gust and Wind Speed. The output of the neural network is the predicted movement. The network could only see the current time series data (not any history). Figure 1 show the accuracy of this prediction.

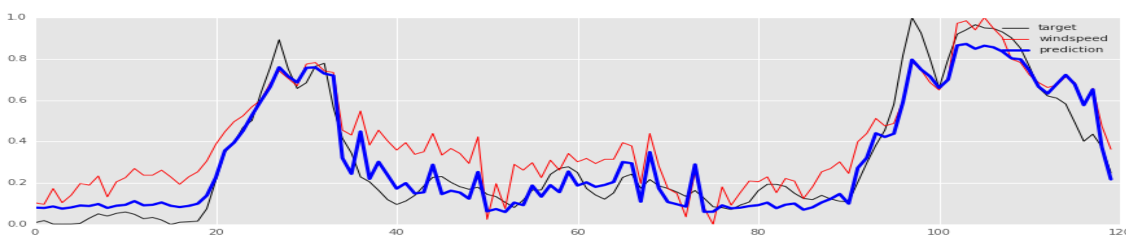


Fig. 1: A graph showing a neural network predicting (blue line) the variation on the x axis of an accellerometer mounted on the aquaculture cage(shown as “target” or a black line in the graph). The red line depicts the wind speed.

As one can see from the figure the network predicts the movement quite precisely, however the wind speed is also highly correlated with the movement of the structure (as can be expected). Thus the neural networks net gain over wind speed is not relatively big, however it adds precision.

3.2 Predicting anchor load

In this experiment we tried to predict the load on the anchors of an aquaculture cage structure based on environmental data gathered from a buoy that is situated close (within 100m) of the

[5] This is a report of which operations could not be performed that day due to unfavourable conditions

cage. The target prediction value is the anchor load in newtons. The prediction was made with a neural network with inputs being 15 data points: current direction and speed at 3 depths, max wind speed, average wind speed, wind direction, significant wave height, wave direction and temperature. The output of the neural network is the predicted load. In the previous experiment the neural network could only see the current data at the current point in the time series. In this experiment the neural network could see the current data as well as the three previous time series points, making it a total of $4 * 15 = 60$ inputs to the neural network. Figure 2 show the accuracy of this prediction.

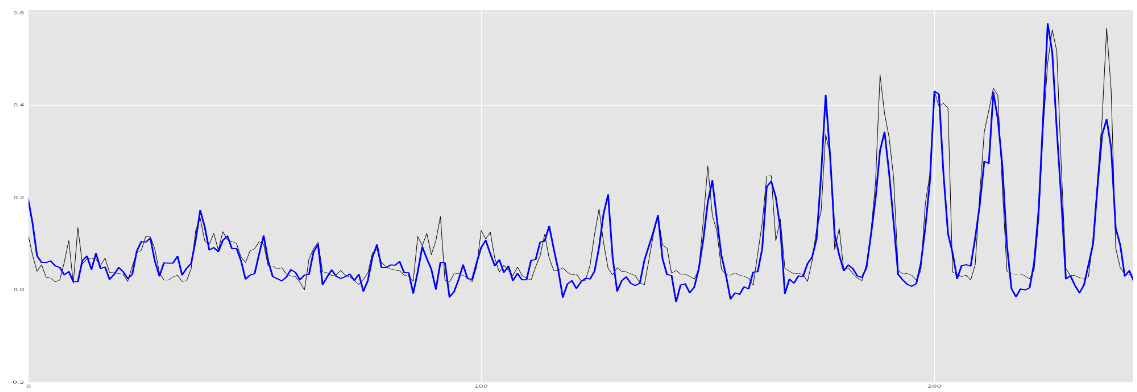


Fig. 2: A graph showing a neural network predicting the anchor load of a aquaculture cage. The blue line depicts the prediction. The black line depicts the actual load.

4 Project plan

In addition to the data listed in Table 1 the project has received a dataset describing the movements of maritime vessels in and out proximity to the aquaculture installations. We have currently combined this with a dataset describing the level of exposure to the environment for each of these sites. In addition we have added weather (wind, waves, temperature, precipitation etc) for each of the events at the relevant sites. The plan is to pick a specific type of boat fishfeed carriers, and then using the time spent at each location to classify whether or not the fishfeed loading operation was successful. We can then use the dataset to try to predict wether such a operation will be successful given the level of exposure and weather forecast. Ideally we would like to use an automated method (likely a ML method) to extract some archetype cases to add to a case-base. This case base could then be employed by a CBR as a part of a decision support system that would be more in line with the type of experience based learning they currently use in the domain.

4.1 Applying CBR

As mentioned earlier we want to apply CBR as the main interface to the user of his DSS. This is because CBR provides a good analogy for the way that this industry learns (experience based learning as opposed to formal learning), and also because CBR is well suited for situations where the data contains instances that are few but have high signal to noise ratio (archetypes); e.g. "Predicted weather conditions are very similar (90%) to a situations where the planned operation failed due to weather conditions." Detecting such these rare instances can most easily be done with using expert knowledge (for verification) and machine learning (for finding previous situations where conditions where far from the average) in tandem. These instances can then be formed into cases where recorded result is shown along with expert input on how to achieve the best result given the conditions (could also be to abort the operation).

Bibliography

[1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

[2] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. A neural probabilistic language model. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

[3] Gerry Bourke, Frank Stagnitti, and Brad Mitchell. A decision support system for aquaculture research and management. *Aquacultural Engineering*, 12(2):111–123, 1993. URL `http://www.sciencedirect.com/science/article/pii/014486099390020C`.

[4] Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

[5] George E. Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Ieee Transactions On Audio Speech and Language Processing*, 20:30–42, 2012. doi: DOI10.1109/TASL.2011.2134090.

[6] Kenneth Alan De Jong. Analysis of the behavior of a class of genetic adaptive systems. 1975.

[7] Douglas H Ernst, John P Bolte, and Shree S Nath. Aquafarm: simulation and decision support for aquaculture facility design and management planning. *Aquacultural Engineering*, 23(1): 121–179, 2000.

[8] Odd Erik Gundersen, Frode Sørmo, Agnar Aamodt, and Pål Skalle. A real-time decision support system for high cost oil-well drilling operations. *AI Magazine*, 34(1):21, 2012.

[9] Halmar Halide, A Stigebrandt, M Rehbein, and AD McKinnon. Developing a decision support system for sustainable cage aquaculture. *Environmental Modelling & Software*, 24(6):694–702, 2009.

[10] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[11] John Henry Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[12] Chris J. Maddison, Aja Huang, Ilya Sutskever, and David Silver. Move evaluation in go using deep convolutional neural networks. In *Under review as a conference paper at ICLR 2015*, 2015.

[13] Minling Pan, Ping Sun Leung, and Samuel G Pooley. A decision support model for fisheries management in hawaii: a multilevel and multiobjective programming approach. *North American Journal of Fisheries Management*, 21(2):293–309, 2001.

[14] SFI-consortium. Exposed aquaculture operations: center for research-based innovation.

[15] David J Spiegelhalter, A Philip Dawid, Steffen L Lauritzen, and Robert G Cowell. Bayesian analysis in expert systems. *Statistical science*, pages 219–247, 1993.

[16] Axel Tidemann, Finn Olav Bjørnson, and Agnar Aamodt. Operational support in fish farming through case-based reasoning. In *Advanced Research in Applied Artificial Intelligence*, pages 104–113. Springer, 2012.

[17] Zhang Xiaoshuan, Fu Zetian, Cai Wengui, Tian Dong, and Zhang Jian. Applying evolutionary prototyping model in developing fidss: An intelligent decision support system for fish disease/health management. *Expert Systems with Applications*, 36(2):3901–3913, 2009.

Maintenance of Case Contents and Adaptation Rules

Brian Schack
Advisor: Dr. David Leake
Indiana University, Bloomington IN 47408, USA
schackb@indiana.edu

Abstract. This research summary outlines and addresses three problems in case-based maintenance on case contents and adaptation rules: 1) how to perform maintenance on divisible cases with non-uniform sizes, 2) how can adaptation knowledge improve the performance of maintenance on structured cases, 3) how to determine and use coverage for adaptation rules. Evaluation showed that, for suitable cases bases, maintenance strategies that subdivide cases and employ adaptation knowledge can outperform per-case strategies. A planned experiment will expand or contract the coverage claimed by adaptation rules and measure the effects on problem-solving performance. The conclusion summarizes research progress to date and areas for further research.

Keywords: case-based reasoning, case-base maintenance, flexible feature deletion, adaptation-guided maintenance, adaptation knowledge

1 Introduction

Case-based reasoning is a method of machine learning for solving problems involving four phases: retrieval, reuse, revision, and retention [1]. Its overall performance depends in no small part on the case base. Even starting with a high-quality case base, the passage of time motivates the need for case-base maintenance. Over time, the system will solve problems and store their solutions in its case base. As these solutions accumulate, they take up storage space, and they take time to search through. The passage of time can also make stored cases in need of revision or obsolete entirely. Over time, even the case representation can change as the system learns more about its domain or its environment changes.

2 Flexible Feature Deletion

My research on flexible feature deletion started by questioning the assumptions for case-based maintenance [3]. First, nearly universally, the evaluations of case-based maintenance strategies assume a uniform size for cases. Although correct for many representations, this assumption does not hold for variable-length feature vectors or more complex structured representations. For example, a case base of films could have different sizes depending on their running times or their numbers of actors and actresses.

This suggests that maintenance strategies employing deletion should consider not only the coverage benefit provided by a case, but also the storage cost of retaining it.

A second assumption states that cases do not permit subdivision of their contents. Normally, a maintenance strategy will either choose to delete or retain an entire case. But if contents permit sub-deletion, abstraction [4], or alteration, then the size of a case base can change independently from the number of cases in it. For example, a case base of medical imagery [5] could delete irrelevant regions or represent them at a lower level of detail.

Dismissing both of these assumptions allows for a classification of different kinds of maintenance strategies depending on how they subdivide the case base: case-bundled, feature-bundled, and unbundled. First, case-bundled strategies delete an entire case including its component features. Second, feature-bundled strategies delete a single feature across all of the cases in the case base. And third, unbundled strategies delete individual a case-feature pair independently of the remainder of the case to which it belongs or the occurrences of the same feature in other cases.

Along those lines, I implemented 11 simple maintenance strategies. The strategies are named according to what they target for deletion first. For example, the Rarest Features strategy deletes features in order from the rarest to the most common. Three hybrid strategies combine pairs of strategies with a 50/50 weighting. The following table compares each of the maintenance strategies:

Strategy	**Type of Bundling**	**Hybrid or Non-Hybrid**
Random Case-Features	Unbundled	Non-Hybrid
Random Cases	Case-Bundled	Non-Hybrid
Largest Cases	Case-Bundled	Non-Hybrid
Least Coverage	Case-Bundled	Non-Hybrid
Most Reachability	Case-Bundled	Non-Hybrid
Random Features	Feature-Bundled	Non-Hybrid
Rarest Features	Feature-Bundled	Non-Hybrid
Most Common Features	Feature-Bundled	Non-Hybrid
Largest Cases / Least Coverage	Case-Bundled	Hybrid
Rarest Features / Least Coverage	Unbundled	Hybrid
Rarest Features / Large Cases	Unbundled	Hybrid

I evaluated the strategies on case bases across three different domains: IMDb films, Congressional bills, and travel agency packages. As always, there is no such thing as a free lunch because of the trade-off between accuracy and size [6]. The question was then which strategy could retain the most competence in spite of deletions. The evaluation showed that the best strategy varied depending on the case base, but for some cases bases, unbundled and feature-bundled strategies could outperform case-bundled strategies.

3 Adaptation-Guided Maintenance

The results for the simple strategies inspired me to ask whether more powerful strategies could achieve a higher level of performance. I looked for sources of knowledge to bring to bear, and adaptation knowledge seemed the most promising [7]. The adaptation phase revises internal case contents in order to make the retrieved solution more suitable to the given problem. And feature-level maintenance also revises case contents, but in this situation, in order to reduce case base size. So, in a sense, both the adaptation and maintenance phases perform adaptations (perhaps even from the same set of possibilities) just with differing goals.

Therefore, I investigated whether maintenance could tie its deletion decisions directly to adaptation knowledge in order to improve on flexible feature deletion. Furthermore, the maintenance strategy could prioritize a deletion of a component within a case according to its recoverability through further adaptations or chains of adaptations. From a different perspective, this approach deletes knowledge overlapping between the case and solution transformation containers [8].

I evaluated this idea in a path planning domain with the goal of finding the shortest path between vertices on a weighted graph representing a route between waypoints on a network of roads. The following table shows the maintenance strategies employed:

Maintenance Strategy	**Lossiness**	**Description**
Shared Component	Lossless	Extract components shared by the solutions of multiple cases into separate cases. Mark gaps for completion during recovery.
Reachability-Based Largest Case	Lossy	Delete cases in order from largest to smallest number of case-features, deleting only recoverable cases.
Largest Case	Lossy	Delete cases in order from largest to smallest number of case-features regardless of recoverability.
Recoverability-Based Random Vertex	Lossy	Delete randomly-chosen case-features from the solutions to cases, deleting only recoverable features.
Random Vertex	Lossy	Delete randomly-chosen case-features from the solutions to cases regardless of recoverability.

Evaluation showed that the Shared Component maintenance strategy retained the most competence as expected because of its classification as lossless. But its compression ability maxed out at about 70% of the size of the uncompressed case base when it could not find any more shared components. Among the lossy strategies, recoverability-based largest case performed next best which showed that the recoverability-based approach can improve competence retention by using adaptation knowledge.

As mentioned earlier, normally competence always decreases with increased compression. However, the results of this experiment surprised me because occasionally adaptation-guided maintenance slightly improved the competence of the system by subdividing cases to make their components accessible to adaptations of limited power – – a phenomenon that I referred to as *creative destruction*.

4 Adaptation Knowledge Coverage

For my current research topic, I considered building on the creative destruction idea. I think I can show theoretically how a formal system could use the same rewriting rules for both adaptation and maintenance, and with suitable rules, creative destruction could have a significant effect. Unfortunately, I haven't found an appropriate domain in which to apply and evaluate this practically.

I settled on the topic of adaptation knowledge coverage. Much research on maintenance has highlighted the importance of the coverage of cases [9], but on the other hand, our field knows comparatively less about how to determine and use coverage for adaptation knowledge. I'm working with a real estate case base consisting of houses for sale with features for their prices, number of bedrooms, square feet, etc. I did not find off-the-shelf adaptation rules for this domain, so I developed a system for learning the rules from pairs of cases (as others have done before me). Together the case base and the learned adaptation rules form an oracle.

Next, I intend to make a copy of the adaptation rules by removing contextual restrictions so that they conflict with one another. I'll eagerly apply rules to the cases and ask the oracle to judge the quality of the derived cases. This will generate triples of case, rule, and quality. From this, I can judge the reliability of the rules and delete the least reliable rules.

Going further, I can look for common features between cases where the same rule applies with a high quality and then restrict the rule to those features. Or alternatively, common features between cases where the same rule applies with a low quality, and then restrict the rule to the negation of those features. The claimed coverage of an adaptation rule could exceed its actual coverage or vice versa. To evaluate this, I'll compare the performance of the oracle, maintained adaptation rules, and unmaintained rules.

5 Further Research

Maintenance necessarily involves a three-fold trade-off between problem-solving competence, problem-solving time, and storage space. Normally, there is no free lunch because reductions in size will also reduce competence. But a lossless maintenance strategy can reduce size to a limited extent without competence reduction, and creative destruction can occasionally even improve competence. Normally, reductions in size mean less cases to search through and therefore reduced retrieval time. But the increased adaptation time to recover a usable solution could cancel out the reduced retrieval time.

Additionally, similarity metrics can involve (perhaps recursively) comparing case components either one-to-one or even many-to-many. Deletion of case components could decrease (or in some comparisons, increase) the time taken by the similarity metric. For example, [10] uses feature reduction to balance the trade-off between the benefit of keeping a feature and the cost of similarity comparisons involving it. In future research, I'd love to explore the different directions of the competence-time-space trade-off on flexible feature deletion and its dependence on the properties of different case bases.

In my research, I used different domains for the different experiments to show broad applicability. But ultimately, I'd like to tie together all of the strategies into a single experiment in order to measure the relative benefit of each separately and together.

6 Conclusion

In conclusion, my research focuses on the maintenance phase of the case-based reasoning cycle. I dismissed the assumptions of uniform size and indivisibility of cases yielding flexible feature deletion strategies. Then, I incorporated adaptation knowledge into these strategies and applied them to structured cases. Next, I intend to continue studying adaptation knowledge especially how to determine the limits of its coverage and how knowledge of coverage for adaptation rules can improve maintenance strategies.

References

1. Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, *7*(1), 39-59.
2. Francis, A. G., & Ram, A. (1993). The utility problem in case-based reasoning. In *Case-Based Reasoning: Papers from the 1993 Workshop* (pp. 160-161).
3. Leake, D., & Schack, B. (2015, September). Flexible feature deletion: compacting case bases by selectively compressing case contents. In *International Conference on Case-Based Reasoning* (pp. 212-227). Springer International Publishing.
4. Bergmann, R., & Wilke, W. (1996). On the role of abstraction in case-based reasoning. *Advances in case-based reasoning*, 28-43.
5. Wilson, D. C., & O'Sullivan, D. (2008). Medical imagery in case-based reasoning. In *Case-Based Reasoning on Images and Signals* (pp. 389-418). Springer Berlin Heidelberg.
6. Lupiani, E., Craw, S., Massie, S., Juarez, J. M., & Palma, J. T. (2013, July). A multi-objective evolutionary algorithm fitness function for case-base maintenance. In *International Conference on Case-Based Reasoning* (pp. 218-232). Springer Berlin Heidelberg.
7. Leake, D., & Schack, B. (2016, October). Adaptation-Guided Feature Deletion: Testing Recoverability to Guide Case Compression. In *International Conference on Case-Based Reasoning* (pp. 234-248). Springer International Publishing.
8. Richter, M. M. (2003). Knowledge containers. *Readings in Case-Based Reasoning. Morgan Kaufmann Publishers*.
9. Smyth, B., & Keane, M. T. (1995, August). Remembering to forget. In *Proceedings of the 14th international joint conference on Artificial intelligence* (pp. 377-382).
10. Floyd, M., Davoust, A., & Esfandiari, B. (2008). Considerations for real-time spatially-aware case-based reasoning: A case study in robotic soccer imitation. *Advances in Case-Based Reasoning*, 195-209.

AToM: An Analogical Theory of Mind

Irina Rabkina[1]

[1] Northwestern University, Evanston, IL 60208, USA
irabkina@u.northwestern.edu

Abstract. Theory of Mind (ToM) is what gives adults the ability to predict other people's beliefs, desires, and related actions, and has been heavily studied in psychology. When ToM has not yet developed, as in young children, social interaction is difficult. Cognitive systems that interact with people on a regular basis would benefit from having a ToM. In this research summary, I propose a computational model of ToM, Analogical Theory of Mind (AToM), based on Bach's [2012, 2014] theoretical Structure-Mapping model of ToM. Completed work demonstrates how ToM might be learned under this model. Future steps include a full implementation and test of AToM.

Keywords: Analogy, Structure Mapping, Theory of Mind

1 Introduction

Humans are inherently social creatures. In fact, it has been suggested that our need for social interaction is responsible for our large brains and incredible language abilities [e.g. Reader and Laland, 2002]. If artificial intelligence systems are to be integrated into our society, then they must share the social capabilities available to us.

Theory of Mind (ToM) is one example of a capability necessary for social interaction. ToM, sometimes referred to as mind reading, is the ability to predict others' desires, beliefs, and other mental states even when they may be different from our own. While some evidence of ToM exists in other highly social animals, such as dolphins and apes [e.g. Krupenye et al. 2016], the extent to which we use and rely on ToM seems to be uniquely human.

Several theories of how ToM is developed and used by humans exist. The philosopher Theodore Bach [2011, 2014] proposed one such theory, based in the Structure-Mapping Theory of analogy [SMT, Gentner, 1983]. This research summary describes a computational cognitive model of ToM, Analogical Theory of Mind (AToM), which is based on Bach's theory. Previous work, which shows how processes which play a role in ToM development can be used to train AToM, is presented. Finally, future directions are discussed.

2 Analogical Theory of Mind (AToM)

AToM is based on the Structure-Mapping Theory of ToM proposed by Bach [2011, 2014]. It is built on top of the Structure-Mapping Engine [SME, Forbus et al. 2016], a computational model of SMT [Gentner, 1983]; the SAGE model of analogical generalization [McLure et al. 2010]; and the MAC/FAC model of analogical retrieval [Forbus et al. 1995]. AToM assumes a long term memory (LTM) of predicate calculus cases that can be retrieved via MAC/FAC. These cases represent memories of life experiences.

When a situation which requires ToM reasoning is encountered, AToM retrieves a relevant case from LTM using MAC/FAC (see Fig. 1). If the retrieved case is a generalized schema, it is applied via analogical mapping as if it were a rule. If the retrieved case is a single event, an interim generalization is created in working memory [Kandaswamy et al. 2014]. While standard interim generalizations are created via SAGE, a slightly different process is involved for AToM's generalizations. Candidate inferences from the retrieved case are projected onto the probe case and, where necessary, portions of the probe case are re-represented. This interim generalization is used for ToM reasoning. AToM then asks for feedback in natural language [using EA-NLU, Tomai and Forbus, 2009]. This is analogous to a person receiving feedback on their reasoning by interacting with others. If the reasoning was correct, AToM uses SAGE to generalize the original probe with the retrieved case, and stores the new generalized case in LTM. Otherwise, it uses MAC/FAC to find a better match (again, given the feedback) and generalizes with the new match. In this way, schemas become more and more generalized, and ToM abilities continue to improve.

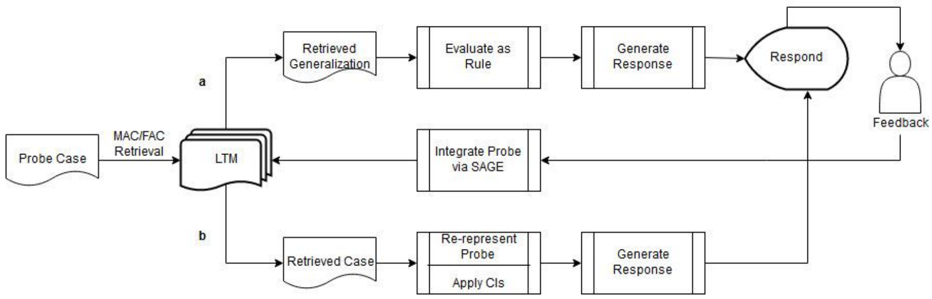


Fig. 1. Process diagram of AToM. Path **a** shows the process when a generalization is retrieved. Path **b** shows the process when a single example case is retrieved.

While AToM is based on Bach's Structure-mapping Theory of ToM [2011, 2014], it differs from the theory in several crucial ways. I will discuss the two biggest differences here. The first major change is to what Bach refers to as the *base representation*, or the case from which reasoning occurs. He suggests that the base representation is formed by re-representing the probe case from the third person into the first person, and adding facts that represent mental state, which are generated by a separate decision-making system. While the interim generalization generated by AToM is analogous to Bach's base representation, the re-representation process is based on a specific retrieved case. The mental state facts, then, are also projected as candidate inferences from the retrieved case, rather than being generated by a separate system.

Another important difference between AToM and Bach's theory lies in the integration of the probe case to LTM. Bach posits that schemas for ToM reasoning are abstracted from simulations. This abstraction happens during construction of the base representation and the comparison between it and the original probe. In AToM, the schemas are instead formed by generalizing the original probe with the retrieved case. In this way AToM builds up its LTM directly from its experiences.

3 Progress to Date

I have completed two computational models of processes involved in ToM learning. These models were used to simulate psychological studies and show results consistent with human data. These results suggest that AToM is a plausible model of ToM reasoning.

3.1 Pretense

Pretend play is ubiquitous throughout childhood. Psychologists believe that it plays a large role in social development in general, and ToM development in particular [Weisberg, 2015]. The mechanisms by which pretense aids with development, however, is an open question. We [Rabkina and Forbus, in prep] suggest that pretense is an analogical process which drives the development of some aspects of analogical reasoning. Because AToM, per Bach, argues that ToM is also analogical, it follows that development of analogical processes will aid ToM development.

Our model of pretense suggests that pretend play relies heavily on analysis of candidate inferences. In the model, when a pretend scenario is encountered, a schema of its real-life equivalent is retrieved. The two are compared via SME, and candidate inferences are projected from the schema to the pretend scenario. Pretend play is successful when the child is able to accept the proper candidate inferences and transform the pretend scenario accordingly. The model successfully replicates the patterns of behavior, including success and failure in pretense, observed in two psychological studies [Fein, 1975; Onishi et al. 2007].

The process by which interim generalizations are formed in AToM is very similar to how they are formed in the pretense model: candidate inferences from the retrieved case must be evaluated and applied to the probe. Thus, it is reasonable that practicing this skill via pretense would improve ToM abilities.

3.2 ToM Training Study

While the pretend play study suggests one mechanism by which ToM might be learned in the wild, psychologists have been able to teach children some aspects of ToM in short intervention sessions. For example, Hoyos et al. [2015] used the repetition break paradigm, described below, to teach children false belief tasks.

In this study, children heard three vignettes. These vignettes were all of the same form: the child is presented with a container (e.g. a crayon box) and asked what they believe is inside. The contents of the container are then revealed. In two of the vignettes, the contents of the box are as expected (e.g. crayons in the crayon box); in the third, they are surprising (e.g. grass in the crayon box).This format is referred to as repetition-break. After the reveal, a new character is introduced, and the child is asked what the character believes is inside the box. In the case where the contents of the box are surprising, the child is expected to answer with the false belief (e.g. the character thinks there are crayons in the box, even though there is actually grass).

From just hearing the three vignettes, children improved significantly on several false belief tasks. Importantly, children who heard vignettes that were highly alignable, that is had high structural similarity, outperformed children who heard vignettes that did not align [Hoyos et al. 2015].

While this alone provides evidence for the role of structure-mapping in ToM development, AToM provides a mechanism by which it may actually happen. In fact, a version of AToM [Rabkina et al. 2017] accurately modeled this task. The model included only a simplified version of the learning steps of AToM: retrieval and integration, along with a reasoning step. Using a simplified-English version of the vignettes and tests used by Hoyos et al. [2015], it replicated the pattern of learning achieved by the children in the study. That is, the model learned false belief tasks from both sets of vignettes, but learned more of them from the vignettes which were highly alignable. Furthermore, the model provided several predictions about ToM in humans.

4 Future Directions

The experiments described above provide evidence that AToM is a plausible mechanism for ToM. However, ToM covers a broad range of phenomena, and a complete model of ToM should be able to model human performance on a variety of tasks. I am currently in the process of identifying additional tasks for testing AToM that would provide a base of evidence that AToM can explain the breadth of ToM reasoning and development in both children and adults.

There are also several areas in which AToM can be improved as a model. For example, the repetition-break study [Hoyos et al. 2015] and our model of it [Rabkina et al. 2017] suggest that surprise plays a role in learning ToM. Incorporating a model of surprise into AToM is a future goal. Furthermore, candidate inference evaluation is important to both AToM and our pretense model [Rabkina and Forbus, in prep]. Developing a cognitively plausible mechanism for these evaluations is also future work.

5 Acknowledgements

This research was supported by the Socio-Cognitive Architectures for Adaptable Autonomous Systems Program of the Office of Naval Research, N00014-13-1-0470.

References

[Bach, 2011] Theodore Bach. Structure-mapping: Directions from simulation to theory. *Philosophical Psychology*, 24(1): 23-51. 2011.

[Bach, 2014] Theodore Bach. A Unified Account of General Learning Mechanisms and Theory-of-Mind Development. *Mind & Language*, 29(3): 351-381. 2014.

[Gentner, 1983] Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7, 155–170. 1983.

[Fein, 1975] Greta G. Fein. A transformational analysis of pretending. *Developmental Psychology* 11.3:291. 1975.

[Forbus et al. 1995] Kenneth D. Forbus, Dedre Gentner, and Keith Law. MAC/FAC: A Model of Similarity-based Retrieval. *Cognitive Science*, 19:141-205. 1995.

[Forbus et al. 2016] Kenneth D. Forbus, Ronald W. Ferguson, Andrew Lovett, and Dedre Gentner. Extending SME to handle large-scale cognitive modeling. *Cognitive Science*, DOI: 10.1111/cogs.12377, pp 1-50. 2016.

[Hoyos et al. 2015] Christian Hoyos, William S. Horton, and Dedre Gentner. Analogical comparison aids false belief understanding in preschoolers. In *Proceedings of the Cognitive Science Society*. 2015.

[Kandaswamy et al. 2014] Subu Kandaswamy, Kenneth Forbus, and Dedre Gentner. Modeling Learning via Progressive Alignment using Interim Generalizations. *Proceedings of the Cognitive Science Society*. 2014.

[Krupenye et al. 2016] Christopher Krupenye, Fumihiro Kano, Satoshi Hirata, Josep Call, and Michael Tomasello. Great apes anticipate that other individuals will act according to false beliefs. Science 354, no. 6308: 110-114. 2016.

[McLure et al. 2010] Matthew McLure, Scott Friedman, and Kenneth Forbus. Learning concepts from sketches via analogical generalization and near-misses. *Proceedings of the 32nd Annual Conference of the Cognitive Science Society (CogSci).* Portland, OR. 2010.

[Onishi et al. 2007] Kristine H. Onishi, Renée Baillargeon, and Alan M. Leslie. 15-month-old infants detect violations in pretend scenarios. *Acta Psychologica* 124.1: 106-128. 200.7.

[Rabkina and Forbus, in prep] Irina Rabkina and Kenneth D. Forbus. An Analogical Model of Pretense. In prep.

[Rabkina et al. 2017] Irina Rabkina, Clifton McFate, Kenneth D. Forbus, and Christian Hoyos. Towards a Computational Analogical Theory of Mind. *Proceedings of the 39th Annual Meeting of the Cognitive Science Society.* London, England. 2017.

[Reader and Laland, 2002] Simon M. Reader and Kevin N. Laland. Social intelligence, innovation, and enhanced brain size in primates. *Proceedings of the National Academy of Sciences*. 99.7:4436-4441. 2002.

[Tomai and Forbus, 2009] Emmett Tomai and Kenneth Forbus. EA NLU: Practical Language Understanding for Cognitive Modeling. *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference*. Sanibel Island, Florida. 2009.

[Weisberg, 2015] Deena S. Weisberg. Pretend play. *Wiley Interdisciplinary Reviews: Cognitive Science* 6.3: 249-261. 2015.

Computer Cooking Contest

Workshop at the
Twenty-Fifth International Conference on
Case-Based Reasoning
(ICCBR 2017)

Trondheim, Norway
June 2017

Nadia A Najjar and David C Wilson (Editors)

Chairs

Nadia A Najjar	University of North Carolina at Charlotte (UNCC), USA
David C Wilson	University of North Carolina at Charlotte (UNCC), USA

Program Committee

David Aha	Naval Research Laboratory, USA
Klaus-Dieter Althoff	DFKI / University of Hildesheim, Germany
Ralph Bergmann	University of Trier, Germany)
Isabelle Bichindaritz	State University of New York at Oswego, USA
Kazjon Grace	University of North Carolina at Charlotte, USA
Ichiro Ide	Nagoya University, Japan
David Leake	Indiana University, USA

Preface

The Computer Cooking Contest aims to attract people working with AI technologies such as case-based reasoning, semantic technologies, search, and information extraction. Also, cooking is fun, particularly when using a computer to design the menu. Since everybody knows something about cooking, people will be curious about how well a computer can cook. Finally, we have all noticed the public's increasing interest in cooking, motivated by the growing awareness that good food is mandatory for good health. Hence, the Computer Cooking Contest provides an opportunity for researchers to explain the benefits of their technologies to everyone.

The Computer Cooking Contest (CCC) is an open competition. All individuals (e.g., students, professionals), research groups, and others are invited to submit software that creates recipes. The primary knowledge source is a database of basic recipes from which appropriate recipes can be selected, modified, or even combined. The queries to the system will include the desired and undesired ingredients. For most of the queries there is no single correct or best answer. That is, many different solutions are possible, depending on the creativity of the software. There is no restriction on the technology that may be used; all are welcome.

This year competition offers four challenges:

- the salad challenge on suggesting salad recipes with a limited set of ingredients and managing the ingredient quantities
- the easy steps challenge on adapting recipes with no restriction on ingredients, but managing the steps
- the mixology challenge on adapting the ingredients of a cocktail recipes with a limited set of ingredients
- and the open challenge on novel ideas and positions on computer cooking

The competition received seven submissions from which six papers were selected as finalists. We are happy to present the contributions of the teams that have been accepted to the Computer Cooking Contest 2017. In "Cooking made easy: On a novel approach to complexity-aware recipe generation" Gilbert Mller and Ralph Bergmann address the easy steps challenge. The approach defines a new complexity-based criterion to be used to guide CookingCAKE's retrieval and adaptation processes that can be tuned as desired against level of query match.

The Taaable team composed of Emmanuelle Gaillard, Jean Lieber and Emmanuel Nauer address the mixology, salad and open challenges in their paper "Adaptation of Taaable to the CCC'2017 Mixology and Salad Challenges, adaptation of the cocktail names". In this adaptation the Taaable as well as the integrated Tuuurbine CBR system uses RDFS for storing domain specific knowledge, which allows comprehensive reasoning strategies. They present a set of approaches to address the different challenges. The first is an approach to adaptation that is used to address constraints arising from a limited set of available ingredients, as well as ingredient quantities, which is applied for the salad and mixology challenges. The second is an approach to name adaptation for cocktail recipes that is applied to the open challenge.

Johnathan Pagnutti and Jim Whitehead contribution, Cooking On The Margins: Probabilistic Soft Logics for Recommending and Adapting Recipes, describes an approach to recipe recommendation and adaptation based on Probabilistic Soft Logics (PSL) that targets the mixology and open challenges for the Computer Cooking Contest.

Kari Skjold, Marthe Oynes, Kerstin Bach and Agnar Aamodt introduce an interactive system in their paper titled "IntelliMeal - Enhancing Creativity by Reusing Domain Knowledge in the Adaptation Process" that targets the open challenge. Their system allows a user to declare desired and undesired ingredients and retrieve relevant recipes from the database. However, it does not stop there. It also generates recipes modified according to the user's declaration. These adapted versions are mixed with the original recipes, filtered, and then presented to the user for manual judgment. It will then be added to the original recipe database if the user judges as appropriate.

In "A Proposed General Formula to Create and Analyze Baking Recipes" Michael Ohene presents a mathematical formula for baking recipes that is, as he argues, capable of identifying unacceptable recipes. The results also produced logical mathematical groupings of baked good recipes. Through the Random Recipe Generator, the author states that it is possible to generate different recipes from characteristic values via ingredient constants.

Christian Zeyen, Gilbert Mller and Ralph Bergmann propose a recipe retrieval method based on Q&A conversations with a user in their paper titled " Conversational Retrieval of Cooking Recipes". The system issues questions to a user based on the workflow derived from the analysis of a recipe. Abstraction of ingredients and operations is performed so that the system can start from asking relatively abstract questions, and then formulating the user's preference (desired and undesired) by traversing up and down the abstractness structure.

The 10th Computer Cooking Contest will be held in conjunction with the 2017 International Conference on Case-Based Reasoning in Trondheim, Norway. A web site with detailed information on the competition and challenges is online at: http://computercookingcontest.com.

Trondheim, Norway	Nadia A Najjar
June 2017	David C Wilson

Cooking made easy: On a novel approach to complexity-aware recipe generation

Gilbert Müller and Ralph Bergmann

Business Information Systems II
University of Trier
54286 Trier, Germany
[muellerg][bergmann]@uni-trier.de
http://www.wi2.uni-trier.de

Abstract. This paper presents an approach to generate easy-to-prepare cooking recipes represented as workflows. A novel complexity-aware generation approach is described that considers various aspects such as preparation time, number of ingredients, and difficulty of preparation to optimize the complexity of the recipe. Based on a user query specifying the desired and undesired ingredients or preparation steps, easy-to-prepare dishes are generated automatically.

Keywords: workflow complexity, workflow adaptation, cooking, process-oriented case based reasoning

1 Introduction

Nowadays, an increasing amount of amateur chefs become fascinated by the world of cooking. Traditional cooking websites support these chefs in finding suitable cooking recipes. However, the recipes need to match several criteria, which sometimes require recipes to be adapted to the individual demands of the user. These demands include contained ingredients, required preparation tools, or dietary restrictions. Thus, several novel approaches have been presented aiming at supporting the user beyond traditional recipe search (e.g., [5,7,3,6]). In certain situations, amateur chefs may prefer easy-to-prepare cooking recipes with a short preparation time, low required cooking skills, or a small amount of ingredients for a variety of reasons.

In this paper we will describe a novel approach that automatically constructs individual and easy-to-prepare cooking recipes based on ingredients and preparation steps specified as desired or undesired. The approach is based on our CookingCAKE framework [10], which will be extended by a new complexity-aware recipe generation. The remainder of this paper is organized as follows: The next section presents the foundations of the CookingCAKE framework. Then, we introduce a complexity assessment for cooking recipes represented as workflows, which will be applied during CookingCAKE's recipe generation. Finally, we present our prototypical implementation for competing in the *Easy Steps Challenge* of the *Computer Cooking Contest*.

2 CookingCAKE

CookingCAKE constructs individual cooking recipes represented as workflows by means of Process-oriented Case-based Reasoning [8]. In a nutshell, CookingCAKE selects the best matching cooking workflow from the workflow repository (case base) and subsequently adapts it according to a query specified by the user.

2.1 Cooking Workflows

In our approach a cooking recipe is represented as a workflow describing the process to prepare a particular dish [13] (see Fig. 1). A cooking workflow $W = (N, E)$ consists of nodes $N = N^T \cup N^D$ and edges $E = E^C \cup E^D$. Nodes of the workflow represent *preparation steps* N^T (also called *tasks*) or *ingredients* N^D (also called *data nodes*). The execution order of preparation steps is defined by *control-flow edges* $E_C \subseteq N^T \times N^T$ and the consumption or production of an ingredient is specified by *data-flow edges* $E_D \subseteq (N^T \times N^D) \cup (N^D \times N^T)$. Furthermore, we enforce that the workflow is executable, which means here that it consists of a single sequence of tasks such that each task $t \in N^T$ consumes ($\exists d \in N^D : (d, t) \in E^D$) and produces (i.e., $\exists d \in N^D : (t, d) \in E^D$) at least one ingredient, respectively. An example cooking workflow for a sandwich recipe is illustrated in Fig. 1.

2.2 Ingredient and Preparation Step Similarity

To support retrieval and adaptation of workflows, the individual workflow elements are annotated with ontological information resulting in a *semantic workflow* [2]. CookingCAKE uses a taxonomy of ingredients to define the semantics of data items and a taxonomy of preparation steps to define the semantics of tasks. These taxonomies are employed for the similarity assessment between tasks and data items. An example ingredient taxonomy is given in Figure 2. A taxonomy is ordered by terms that are either a generalization or a specialization of a specific other term within the taxonomy, i.e., an inner node represents a generalized term that stands for the set of most specific terms below it. For example, the generalized term $vegetarian$ in the illustrated taxonomy

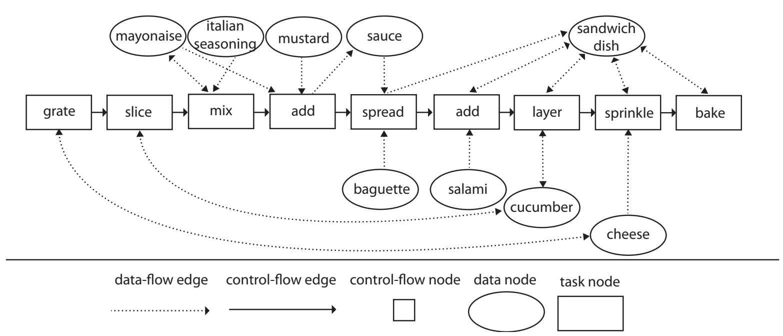


Fig. 1. Example of a block-oriented cooking workflow

stands for the set $\{potatoes, rice, noodles, \ldots\}$. Inner nodes in generalized workflows represent that an arbitrary ingredient from the set of its specializations can be chosen.

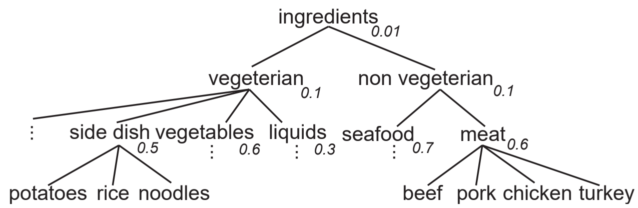


Fig. 2. Example of an ingredient taxonomy

In our previous work, we developed a semantic similarity measure for workflows that enables the similarity assessment of a case workflow W_c w.r.t a query workflow W_q [2], i.e. $sim(W_c, W_q)$. Each query workflow element $x_q \in W_q$ is mapped by the function $m : W_q \rightarrow W_c$ to an element of the case workflow $x_c \in W_c$, i.e., $x_c = m(x_q)$. The mapping is used to estimate the similarity between the two workflow elements utilizing the taxonomy, i.e., $sim(x_q, x_c)$. The similarity of preparation steps or ingredients reflects the closeness in the taxonomy and further regards the level of the taxonomic elements. In general, the similarity is defined by the attached similarity value of the least common ancestor, e.g., $sim(beef, pork) = 0.6$. If a more general query element such as *meat* is compared with a specific element below it, such as *pork*, the similarity value is 1. This ensures that if the query asks for a recipe containing meat, any recipe workflow containing any kind of meat is considered highly similar. All the similarity values of the mappings are then aggregated to estimate an overall workflow similarity.

2.3 Workflow Query Language

CookingCAKE uses the query language POQL [12] to capture desired and undesired ingredients or preparation steps of a cooking workflow as query q. The ability to specify preparation steps is useful as certain tools might not be available or their usage is desired (e.g., oven). Let $q_d = \{x_1, \ldots, x_n\}$ be a set of desired ingredients or preparation steps and $q_u = \{y_1, \ldots, y_n\}$ be a set of undesired ingredients or preparation steps, respectively. A query q is then defined as $(x_1 \wedge \ldots \wedge x_2) \wedge \neg y_1 \wedge \ldots \wedge \neg y_n$. POQL further enables the specification of generalized terms, i.e., if a vegetarian dish is desired, this can be defined by $\neg meat$. The query q is used to guide retrieval, i.e., to search for a workflow which at best contains all desired elements but no undesired element. Based on the query q the not matching elements can be identified, enabling to determine the elements to be deleted or added to the retrieved workflow during the subsequent adaptation stage. The query fulfillment of a workflow W for a query q is defined as the similarity between the desired ingredients/preparation steps as well as the workflow W and the number of undesired ingredients/preparation steps not contained in W according to the workflow similarity (see Sec. 2.2) in relation to the size of the query (see Formula 1).

$$QF(q,W) = \frac{\sum_{x \in q_d} sim(x, m(x)) + |\{y \in q_u | sim(y, m(y)) \neq 1\}|}{|q_d| + |q_u|} \quad (1)$$

Consequently, similar desired ingredients or preparation steps increase the query fulfillment, while matching undesired ingredients or preparation steps reduce the query fulfillment between the POQL query and the workflow.

2.4 Recipe Construction

Based on the defined POQL query, CookingCAKE constructs a workflow automatically by retrieving the best matching workflow from the repository (*case base*) and adapting it according to the query fulfilment. Consequently, the adaptation process of CookingCAKE aims at adding missing desired ingredients/preparation steps and at removing undesired contained ingredients/preparation steps. In a nutshell, the adaptation process uses three different adaptation methods that are subsequently executed. First, entire components of the cooking dish such as the *sandwich sauce* or *sandwich topping* are replaced by matching components from other recipes [9]. Next, adaptation is performed by use of operators that define possible and valid modifications on the cooking workflows. Finally, the cooking recipes are adapted by replacing single ingredients and preparation steps by means of the specified taxonomy, assuming that similar terms can most likely be replaced with each other [11]. In all approaches, adaptation of a workflow is performed by chaining several adaptation steps $W \xrightarrow{\alpha_1} W_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} W_n = W'$, which iteratively transforms the retrieved workflow W towards an adapted workflow W'. This process solves an optimization problem aiming at maximizing the specified criterion, which is so far implemented by the query fulfillment. Thus, the recipe construction is a search process with the goal to achieve an adapted workflow with the highest query fulfillment possible. The overall recipe construction process ensures the syntactical correctness of the workflows, i.e., that the workflows are executable. More detailed information on the construction process of CookingCAKE can be found in the corresponding publication [10].

In the next section, we introduce a new criterion for the retrieval and adaptation process that considers the complexity of workflows. Thus, retrieval as well as the adaptation become complexity-aware and aim at optimizing the constructed workflow with regard to the new defined criterion during recipe construction.

3 Complexity Assessment

In the literature various approaches to asses the complexity of workflows exist (see [4]). In this approach, we rather focus on a domain-specific complexity measure for cooking workflows. During recipe construction, this complexity criterion is considered to generate easy-to-prepare recipes automatically. We assume that the complexity of a recipe is less focused on one single feature, but is composed by several criteria. Thus, we deploy a complexity measure that covers five different indicators for determining the complexity of the recipe (see Table 1).

Table 1. complexity criteria

critera description	**criteria measure**
number of ingredients	$\frac{\lvert N^D \rvert}{max\{\lvert N_1^D \rvert,\ldots,\lvert N_n^D \rvert\}}$
number of preparation steps	$\frac{\lvert N^T \rvert}{max\{\lvert N_1^T \rvert,\ldots,\lvert N_n^T \rvert\}}$
complexity of ingredient processing	$1 - \frac{2 \cdot \lvert N^T \rvert}{\lvert E^D \rvert}$
complexity of preparation steps	$\frac{\sum\limits_{t \in N^T} taskComplexity(t)}{\lvert N^T \rvert}$
duration of preparation	$\frac{preparationTime(W)}{max\{preparationTime(W_1),\ldots,preparationTime(W_n)\}}$

The first two criteria measure basic complexity properties, i.e., the *number of preparation steps* as well as the *number of ingredients* in the particular cooking workflow $W = (N, E)$. Both measures are normalized by the highest amount of ingredients or preparation steps contained in the workflows from the workflow repository. Consequently, cooking workflows with more ingredients or more preparation steps are assumed to be more complex. Furthermore, the *complexity of preparation steps* as well as the *complexity of ingredient processing* represent two additional complexity criteria. The complexity measure for *ingredient processing* considers the average amount of ingredients consumed and produced by the preparation steps, which assigns a high complexity value to those workflows in which the preparation steps N^T consume and produce a large amount of ingredients[1] E^D. In contrast, for computing the *complexity of preparation steps* each task t in the taxonomy (see Sec. 2.2) is annotated by an estimated task complexity value $taskComplexity(t) \in [0, 1]$. As an example, the preparation step *blanche* is considered to be more complex than the preparation step *mix*. The criterion is then defined as the average complexity of the preparation steps in the workflow W. Finally, the duration for preparing a particular dish is also a factor affecting the complexity. Therefor, also approximated execution times $taskPreparationTime(t) \in \mathbb{N}$ are annotated to each task t in the taxonomy. Here, for example, *baking* is annotated by a long execution time, while *season* is considered as a rather short preparation step. The *duration of preparation* for a workflow W is then heuristically measured by aggregating the execution times of the preparation steps, i.e., $preparationTime(W) = \sum_{t \in N^T} taskPreparationTime(t)$. To assess the corresponding complexity, this value is normalized in relation to the workflows from the repository as defined in Table 1.

Each of these five complexity measures determines a complexity value within the interval $[0, 1]$. Based on these measures, we constructed an overall complexity measure $complexity(W) \rightarrow [0, 5]$ which adds up all complexity criteria to a single value. The

[1] Please note that each task in a workflow consumes and produces at least one ingredient, respectively (see Sec. 2.1)

overall complexity measure specifies the corresponding difficulty level of the recipe preparation and distinguishes between *very easy* ($[0, 1[$), *easy* ($[1, 2[$), *medium* ($[2, 3[$), *difficult* ($[3, 4[$) and *very difficult* ($[4, 5]$).

$$QF_{\text{complexity}}(q, W) = \alpha \cdot QF(q, W) + (1 - \alpha) \cdot (1 - complexity(W)/5) \quad (2)$$

Based on this overall complexity measure, we defined a new complexity-aware query fulfilment measure $QF_{\text{complexity}}(q, W) \to [0, 1]$ (see Eq. 2) for the retrieval and adaptation process. It replaces the query fulfillment measure specified in formula 1, thus considering complexity as well. Both criteria may be weighted by a parameter $\alpha \in [0, 1]$. The workflow construction process of CookingCAKE as described in Section 2.4 then aims at optimizing the constructed workflow with regard to this new criterion. Please note that this is a multi-objective optimization problem and thus the adaptation may not be able to maximize the query fulfillment and to reduce the complexity of the workflow at the same time.

4 Computer Cooking Contest: Easy steps challenge

We created a new user interface for the CookingCAKE system in order to address the *Easy Steps Challenge* of the *Computer Cooking Contest*, which applies the previously described complexity assessment. A running prototype of the implementation is available under (`http://cookingCAKE.wi2.uni-trier.de/complexity`), which uses a workflow repository of 61 sandwich recipes manually modelled from various Internet sources (e.g., sandwich recipes on WikiTaaable [1][2]). The employed taxonomies of preparation steps and ingredients (see Sec. 2.2) are based on the WikiTaaable ontology and were manually annotated with similarity, preparation time, and task complexity values.

The query of CookingCAKE involves desired and undesired ingredients as well as desired and undesired preparation steps. An example query (`http://cookingCAKE.wi2.uni-trier.de/complexity?d=cherry%20tomato|salmon&u=cheese`), generates a *salmon* and *cherry tomato* recipe without using any kind of *cheese*. CookingCAKE then selects the best matching workflow from the repository and subsequently adapts it according to the novel criterion $QF_{complexity}(q, W)$. Thus, the system tries to maximize the query fulfilment on the one hand and on the other hand aims at reducing the complexity of the workflow to generate an appropriate easy-to-prepare recipe for an amateur chef. The result page of the novel CookingCAKE interface also displays the estimated difficulty of preparation, the computed duration time as well as the single complexity values (see Sec. 3) for the constructed recipe.

To evaluate our new complexity-aware approach for recipe construction, we generated 61 queries automatically. More precisely, for each workflow W, a corresponding query was constructed by selecting the most similar workflow W' from the repository and by determining the difference between the two workflows. The constructed query considers workflow elements as desired that are only contained in the workflow W while the elements only contained in workflow W' are considered as undesired. At

[2] http://wikitaaable.loria.fr

most 4 randomly selected ingredients and 2 preparation steps are determined as desired or undesired respectively. For each of the queries we performed a leave-one-out test, i.e., the corresponding workflow was removed from the repository. Then, we executed the recipe generation process with the standard approach as well as the complexity-aware approach. For the complexity-aware recipe construction we chose the parameter $\alpha = 0.5$ to consider the query fulfillment and the complexity in equal shares. For both approaches, we measured the query fulfillment, the complexity, and the combined complexity-aware criterion of the retrieved as well as of the adapted workflow.

Table 2. evaluation results

	query fulfillment	complexity	combined	computation time
standard retrieval	0.83	0.43	0.70	1.15 s
standard adaption	0.92	0.48	0.72	18.73 s
complexity-aware retrieval	0.75	0.28	0.74	1.42 s
complexity-aware adaption	0.87	0.29	0.79	9.49 s

The evaluation results illustrated in Table 2 clearly show that already during complexity-aware retrieval, a less complex workflow is selected. Furthermore, the computation time of the subsequent adaptation stage is significantly decreased[3]. The most important observation, however, is that with the new complexity-aware approach, the final complexity is significantly reduced (-40%), while the query fulfillment is only slightly decreased (-5%). Altogether it can be concluded that the complexity-aware approach presented in this paper enables the individual construction of easy-to-prepare cooking recipes with a low preparation complexity.

5 Conclusions and Future Work

This paper presents a new approach to generate easy-to-prepare cooking recipes based on cooking workflows. The new approach considers a query specified by the user to automatically generate a cooking workflow matching the users demands and further considers the complexity of the cooking workflow as an additional criterion. The complexity measure is composed of several criteria including the number of ingredients, the preparation time and the complexity of preparation steps.

In future work we aim at providing an interface for choosing the desired recipe complexity. Furthermore, the complexity assessment will be improved and evaluated by comparing various complexity measures. Finally, we will investigate several other factors that could be considered during the construction of recipes such as nutritions and dietary restrictions.

[3] The adaptation time depends on the size of the workflow, which is usually smaller, if the workflow is less complex.

Acknowledgements. This work was funded by the German Research Foundation (DFG), project number BE 1373/3-3.

References

1. Badra, F., Bendaoud, R., Bentebibel, R., Champin, P., Cojan, J., Cordier, A., Desprès, S., Jean-Daubias, S., Lieber, J., Meilender, T., Mille, A., Nauer, E., Napoli, A., Toussaint, Y.: TAABLE: text mining, ontology engineering, and hierarchical classification for textual case-based cooking. In: Schaaf, M. (ed.) ECCBR 2008, Trier, Germany, Workshop Proceedings. pp. 219–228 (2008)
2. Bergmann, R., Gil, Y.: Similarity assessment and efficient retrieval of semantic workflows. Inf. Syst. 40, 115–127 (Mar 2014)
3. Bridge, D., Larkin, H.: Creating new sandwiches from old. In: Nauer, E., Minor, M., Cordier, A. (eds.) Procs. of the Seventh Computer Cooking Contest (Workshop Programme of the Twenty-second International Conference on Case-Based Reasoning). pp. 117–124 (2014)
4. Cardoso, J.S., Mendling, J., Neumann, G., Reijers, H.A.: A discourse on complexity of process models. In: Eder, J., Dustdar, S. (eds.) Business Process Management Workshops, BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4-7, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4103, pp. 117–128. Springer (2006), http://dx.doi.org/10.1007/11837862_13
5. Gaillard, E., Lieber, J., Nauer, E.: Improving ingredient substitution using formal concept analysis and adaptation of ingredient quantities with mixed linear optimization. In: Kendall-Morwick, J. (ed.) Workshop Proceedings from (ICCBR 2015), Frankfurt, Germany. CEUR Workshop Proceedings, vol. 1520, pp. 209–220. CEUR-WS.org (2015)
6. Grace, K., Maher, M.L., Wilson, D.C., Najjar, N.A.: Combining CBR and deep learning to generate surprising recipe designs. In: Goel, A.K., Díaz-Agudo, M.B., Roth-Berghofer, T. (eds.) ICCBR 2016, Atlanta, GA, USA, Proceedings. Lecture Notes in Computer Science, vol. 9969, pp. 154–169. Springer (2016), http://dx.doi.org/10.1007/978-3-319-47096-2_11
7. Keppler, M., Kohlhase, M., Lauritzen, N., Schmidt, M., Schumacher, P., Spät, A.: Goethe-Shaker - Developing a rating score for automated evaluation of cocktail recipes. In: ICCBR Workshop Proceedings. pp. 101–108 (2014)
8. Minor, M., Montani, S., Recio-Garca, J.A.: Process-oriented case-based reasoning. Information Systems 40(0), 103 – 105 (2014)
9. Müller, G., Bergmann, R.: Workflow Streams: A Means for Compositional Adaptation in Process-Oriented Case-Based Reasoning. In: Proceedings of ICCBR 2014. Cork, Ireland (2014)
10. Müller, G., Bergmann, R.: CookingCAKE: A framework for the adaptation of cooking recipes represented as workflows. In: Kendall-Morwick, J. (ed.) Workshop Proceedings from (ICCBR 2015), Frankfurt, Germany. CEUR Workshop Proceedings, vol. 1520, pp. 221–232. CEUR-WS.org (2015)
11. Müller, G., Bergmann, R.: Generalization of Workflows in Process-Oriented Case-Based Reasoning. In: 28th International FLAIRS Conference. AAAI, Hollywood (Florida), USA (2015)
12. Müller, G., Bergmann, R.: POQL: A New Query Language for Process-Oriented Case-Based Reasoning (2015), to be submitted to LWA 2015 Trier, Germany
13. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of procedural knowledge from the web. In: Workshop Proceedings: WWW'12. Lyon, France (2012)

Conversational Retrieval of Cooking Recipes

Christian Zeyen, Gilbert Müller, and Ralph Bergmann

Business Information Systems II
University of Trier
54286 Trier, Germany
[zeyen][muellerg][bergmann]@uni-trier.de,
http://www.wi2.uni-trier.de

Abstract. This paper presents a new approach for exploring a collection of cooking recipes represented as cooking workflows by means of a conversation. Users are guided through the search process by answering posed questions. Thus, they are not required to formulate queries and they do not need to browse a recipe collection by hand. Questions involve ingredients and cooking activities contained in the workflows. The approach is implemented in our CookingCAKE system, extending it with a dialog component.

Keywords: Recipe Retrieval, Conversational Retrieval, Workflows

1 Introduction

Nowadays, numerous cooking recipes are available online and various search engines support users in finding suitable recipes. Beside providing a keyword-based search, some engines also support an ingredient-based search by asking the user to specify desired and undesired ingredients. However, in practice, amateur chefs may only have a vague idea of their desired dish or they lack detailed knowledge about required ingredients and thus have difficulties in providing a precise query. In 2010, Yummly[1] launched the first semantic search platform for food and recipes. By capturing the semantics of recipe descriptions and ingredients, Yummly is able to handle more vague queries such as general terms in a keyword-based search. For example, if the user starts a search with the keyword *meat*, Yummly initiates a dialog asking the user which kind of meat she would like. Then, further questions are posed concerning various properties such as the desired type of dish, preparation time, nutritional preferences, and additional ingredients.

This paper follows a similar approach and provides a method to conduct a conversation with the user to find desired cooking recipes. We focus on structural features of recipes thus representing them as workflows. More precisely, in addition to considering the occurance of ingredients and preparation steps, we also analyze their dependencies. Based on this information, questions concerning the further processing of desired ingredients can be posed. Thus, the ingredient-based search capabilities provided in typical search engines for recipes are extended by this approach. Moreover, we investigate how such features can be constructed automatically from the underlying workflow

[1] www.yummly.com

repository and we propose a respective question selection strategy for the conversation. We consider a recipe search as a problem solving process in which the problem description specifies the user's preferences of a desired dish and a possible solution is a recipe describing its preparation. We apply the methodology of conversational case-based reasoning (CCBR) [1,2], which particularly focuses on the interactive nature of problem solving. CCBR approaches include methods which incrementally elicit the relevant features of the target problem in an interactive dialog, often with the aim of minimizing the communication effort for the user. The basic assumption behind CCBR is that guided question answering requires less domain expertise than providing detailed queries from scratch. To apply CCBR with cooking workflows, we combine CCBR with process-oriented case-based reasoning (POCBR) [5], which usually deals with cases as workflows or process descriptions expressing procedural experiential knowledge. Consequently, we propose a new conversational POCBR approach [10], called C-POCBR, for the retrieval of cooking workflows. We implemented the approach in our CookingCAKE system [6], which is part of the CAKE framework[2], extending it with a dialog component. CookingCAKE is a POCBR system for retrieving and adapting[3] cooking workflows based on a user-defined query specifying desired and undesired ingredients and preparation steps.

In the following, section 2 briefly introduces the representation and querying of cooking workflows before section 3 describes our C-POCBR approach. Section 4 concludes the paper and briefly discusses future work.

2 Cooking Workflows

In our approach a cooking recipe is represented as a workflow describing the process to prepare a particular dish [9,3] (see Fig. 1). Cooking workflows consist of a set of *preparation steps* (also called *task nodes*) and a set of *ingredients* (also called *data nodes*) shared between its tasks. Task nodes are linked by *control-flow edges* defining the execution order. This forms the *control-flow*. Task nodes, data nodes, and relationships (represented by *data-flow edges*) between the two of them form the *data-flow*. To each

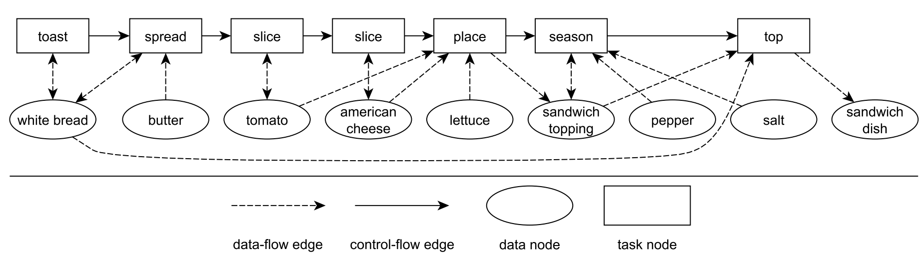


Fig. 1. Example of a Cooking Workflow

[2] See cake.wi2.uni-trier.de
[3] However, the dialog component does not yet consider the available adaptation methods.

node a semantic label is assigned, which is structured hierarchically in a data taxonomy of ingredients or a task taxonomy of cooking steps. Thereby, workflows can be generalized regarding their semantic labels [7]. Generalized workflows provide a more general description and thus stand for a set of more specific workflows. For example, the workflow in Figure 1 can be generalized by generalizing the ingredient *american cheese* to the more general ingredient *cheese* from the data taxonomy.

In order to retrieve cooking workflows with case-based reasoning, the user's preferences must be specified in a query, which in turn must be evaluated against the available workflows. For this purpose, we proposed a process-oriented query language (POQL) [8] and a similarity measure [10] to determine the best-matching workflows for a given POQL query. In a nutshell, POQL consists of two parts. In a *query part*, the user can specify workflow fragments representing properties the searched recipe should fulfill. In an additional *restriction part*, the user can define undesired situations, e.g., unwanted ingredients, which should be avoided.

3 A Conversational Retrieval Approach

To facilitate the elaboration of a POQL query for workflow retrieval, our approach guides the user through the query process with a sequence of questions about her preferences. The more questions are answered, the more knowledge about desired and undesired properties is available, which is stored in an internal POQL query. A major focus is put on the automatic creation of questions to avoid that they need to be specified manually. For this purpose, we consider workflow fragments as characteristic properties of a workflow, which we refer to as features. The basic idea is to extract features from the workflows stored in the repository (case base) automatically, which are then used as the subject of questions. In order to conduct efficient conversations, we rank features by their ability to distinguish workflows from one another. Furthermore, identified relations between features enable to generate coherent follow-up questions and to infer irrelevant features based on already answered questions.

3.1 Features of a Cooking Workflow

In principle, a feature can be any fragment of a cooking workflow. In a workflow, the smallest possible feature consists of a single workflow item. This can be a single node such as a data or a task node. More complex features can be created by extracting partial workflows. To derive questions on a more general level of detail, we apply a generalization algorithm [7], which generalizes semantic labels based on the task and data taxonomies. The generalization produces a generalized workflow W^* from the original workflow W, from which more general features can be extracted. We extract and annotate two different kinds of features for each workflow W in the case base:

- *specific feature nodes* and *generalized feature nodes*, i.e., single nodes from W and single nodes for all generalizations within the taxonomy up to the respective node in the generalized workflow W^*
- *specific feature workflows* and *generalized feature workflows*, i.e., partial workflows (consisting of more than one node) from W and W^*, respectively

A *feature workflow* W_d describes structural properties of a workflow W regarding a particular data item d from W. It consists of all (and at least one) tasks connected to d and which are connected by control-flow edges. Moreover, W_d additionally comprises all data items that are connected to those tasks. For instance, regarding the cooking workflow depicted in Figure 1, a feature workflow constructed for the ingredient *white bread* contains the associated processing steps *toast* and *spread*. It further comprises the ingredient *butter*, which is required for the task *spread*.

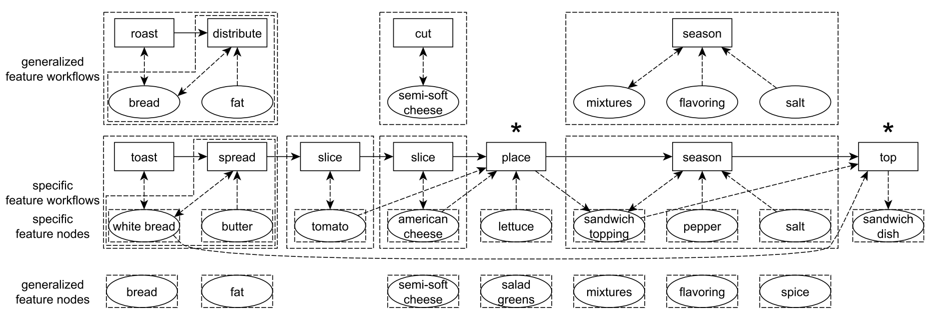


Fig. 2. Examples of a Workflow's Features

Figure 2 exemplifies all features (see dotted rectangles) extracted from the cooking workflow depicted in Figure 1 . The specific workflow is depicted in the middle of the figure. Related features (such as specific and generalized features) are arranged near one another. For instance, the specific feature node *pepper* is related to the generalized feature node *flavoring*. Based on the taxonomy, an additional generalized feature node *spice* laying inbetween those two is extracted as well.

With respect to the cooking domain, we applied some domain-specific restrictions. For the feature extraction, we omit single task nodes as they are mostly of no relevance when considered on their own. In addition, to obtain easy-to-understand feature workflows, we exclude tasks (marked with "$*$") that produce new data by consuming other data.

In a second step all extracted features are sorted in descending order by their ability to distinguish workflows from one another. By this means, we reduce the length of a conversation. We adopt the simVar measure by Kohlmaier et al. [4], which utilizes the similarity variance as a ranking criterion. It estimates the variance of the similarity of the most similar workflows assuming that the value of the respective feature in the query is known (see [10] for more details).

In the next step, relations between features are analyzed. For each feature f all related features are determined. The set of related features of a feature f contains those features g that share a common partial workflow with f which is either a generalization of f or g. Related features can be differentiated by their number of nodes and by their generality of nodes. A feature may have related features that are *larger*, *equally large*, or *smaller* as well as related features which are *more specific*, *equally specific*, or *more general*. For example, for the feature workflow $f_1 = \{$*slice, ham*$\}$, the related feature $g_1 = \{$*cut, meat*$\}$ is *more general* and *equally large* while the feature $g_2 = \{$*parma-ham*$\}$ is *more specific* and *smaller*.

3.2 Questioning Strategy

In order to obtain the user's preferences most suitable to determine the best matching cooking workflows, i.e., the candidate workflows, a respective questioning strategy is required. The dialog component iteratively creates and displays questions until the user selects a desired workflow. With each question answered by the user, the set of candidate workflows, which encompasses the whole case base at the beginning of a conversation, is reduced. The dialog starts with an empty query and the set of candidate features, i.e., relevant features to be asked in a question, comprises the full set of features.

In the main loop, the dialog component selects a question based on the candidate features. The selection process considers the previously answered questions as well as the ranking and relationships of features. Each question involves one or in certain cases several candidate features. We provide three major types of questions, which are depicted in Table 1. Based on the ranking of the candidate features, the subject matter of a question is determined. If the user answers that the suggested feature is desired, specific follow-up questions are selected in the subsequent iterations. Those follow-up questions are derived from related features and aim at further refining the previous question asked. An example is given in Table 1, which presents a sequence of three questions (Q) including the possible answers (A) and the user-selected answers (marked with a box).

Table 1. Question Sequence in a Conversation

Order	Question Type	Subject Matter	Example
1.	initial feature question (FQ)	highest ranked feature	Q: Is $\{meat\}$ a desired feature? A: $\boxed{\text{desired}}$, undesired, irrelevant
2.	follow-up specialization question (SQ)	more specific feature(s)	Q: Is there a suitable specialization for $\{meat\}$? $\{poultry\}, \{ham\}, \boxed{\{\text{chicken}\}}, \ldots$ A: $\boxed{\text{apply}}$, select undesired feature(s), irrelevant
3.	follow-up enlargement question (EQ)	larger feature(s)	Q: Is there a suitable enlargement for $\{chicken\}$? $\boxed{\{\text{shred, chicken}\}}, \boxed{\{\text{chop, chicken}\}}, \ldots$ A: apply, $\boxed{\text{select undesired feature(s)}}$, irrelevant

At the beginning of a conversation the highest ranked feature from the candidate features is suggested in a feature question (FQ). This type of question is not related to previously suggested features and it will be asked as long as the user selects the suggested feature as irrelevant or undesired. In the example, the feature *meat* is the subject matter of the first question and it is selected as *desired* by the user.

If a feature in a FQ is selected as desired, a first follow-up question, i.e., a specialization question (SQ), is posed suggesting one or (if available) several equally large but more specific features. The user can choose a specialization, select specializations as undesired, or mark all specializations as irrelevant. This type of question is repeated as long as further specializations exist and are desired by the user. In the example, the user chooses *chicken* as her desired type of *meat*.

Following the SQs, an enlargement question (EQ) is displayed to the user that suggests, if available, larger features than the previously selected and/or specialized feature. Just as in SQ, the user has three different options: choose an enlargement, select enlargements as undesired, or mark all enlargements as irrelevant. In the given example, we assume that the user does not like *shredded* or *chopped chicken* and thus selects those features as undesired. If no more EQs are available, the next initial FQ is selected, addressing a new and potentially unrelated subject matter.

When the set of candidate features is updated due to an ignored or answered question, irrelevant features are inferred based on the relations between features. If a question is marked as irrelevant, all the related features (e.g., more specific and larger features) are marked as irrelevant, too. If suggested features are selected as undesired, they are added to the restriction part of the current query and related irrelevant features are no longer considered as candidate features, to prevent the system from repetitively asking the user what she does not like. If a feature is marked as desired, also related features such as more general features are removed from the set of candidate features. If a user chooses a specialization or an enlargement, the target feature that is already present in the query is replaced with the new feature.

3.3 Conversation with CookingCAKE

Based on the extracted features and the questioning strategy, the conversation is conducted in the dialog component of CookingCAKE[4]. The graphical user interface is illustrated in Figure 3. It consists of three displays suggesting the best matching workflow (upper part of figure), showing a question (middle of figure), and summarizing the current query (lower part of figure). Figure 3 presents a progressed state in a conversation in which some preferences are already obtained from the user and specified in the internal query by the system. In the given example, the current query contains *firm cheese* as desired and *meat* as undesired. The question displayed is a follow-up question targeting the further refinement of the current query. In the example, the question suggests alternative processing steps for *firm cheese*. The user has two options to react:

1. *Ignore the question:* In this case, the features being subject of the question as well as related features are ignored and the next best question is displayed.
2. *Answer the question:* Causes the system to extend the query and to perform a similarity-based retrieval on the current set of candidate workflows. The workflow with the highest similarity is displayed to the user.

In the upper part of the user interface, a solution workflow best fulfilling the current query is suggested to the user. With respect to this suggestion, the user has two additional options to react:

[4] Online demo available at `cookingcake.wi2.uni-trier.de/conversation`

1. *Ignore the suggestion:* The user can actively ignore the suggested workflow, which causes the system to exclude it from the solution candidates and to trigger a new retrieval for the next best workflow.
2. *Select the suggestion:* In this event, the conversation terminates successfully.

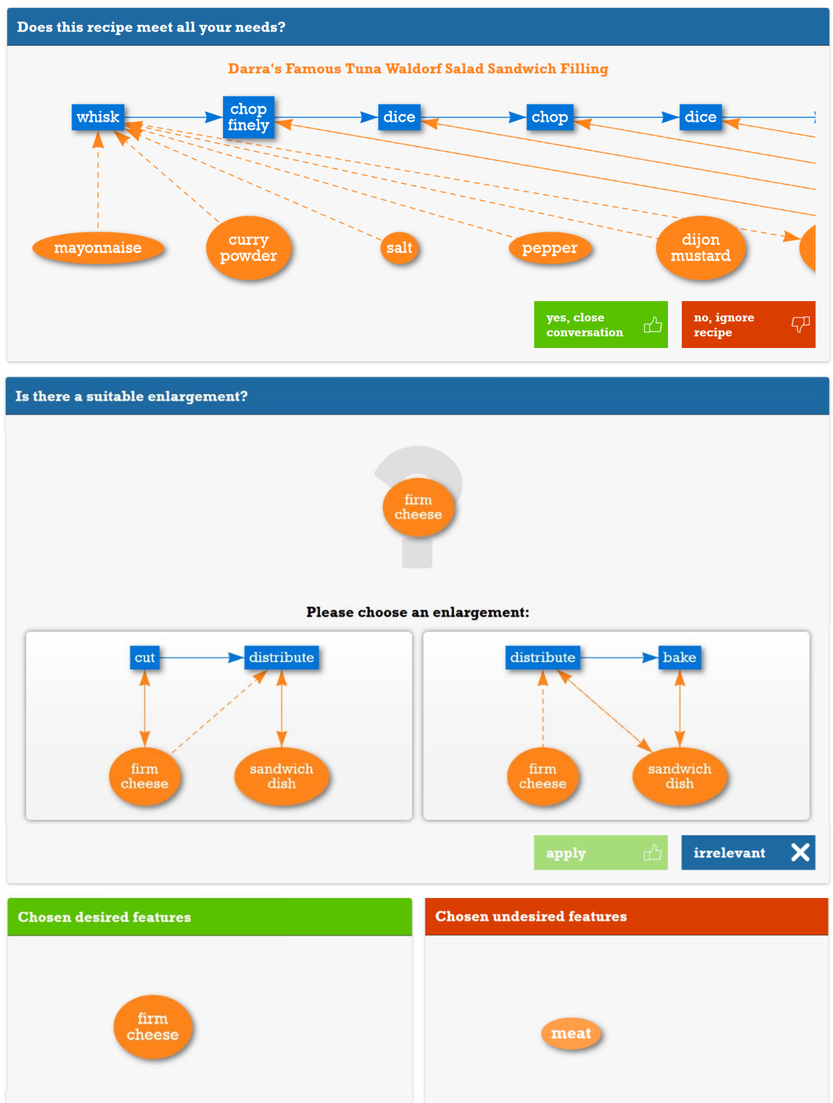


Fig. 3. Dialog Component of CookingCake

4 Conclusions and Future Work

We presented an approach to retrieve cooking workflows by means of an interactive dialog with users. To save effort for defining suitable questions, a method for the automatic creation of questions based on extracted features was described. We recently showed in an experimental evaluation that those features are meaningful subjects of questions and that they are suitable to distinguish workflows from one another [10]. Furthermore, our results indicate that the conversational approach has the potential to reduce the retrieval time and thus is able to reduce the communication effort for users.

In future work we plan to extend the presentation and explanation of workflows and features. For the sake of simplicity, we used a simplistic representation of cooking recipes and considered basic features, which could be extended in the future. Also, future work should investigate how adaptability of workflows can be considered during a conversation. By this means, interactive retrieval could be combined with interactive adaptation to provide more diverse and customized cooking workflows for users.

Acknowledgments. This work was funded by the German Research Foundation (DFG), project number BE 1373/3-3.

References

1. Aha, D.W., Breslow, L.A., Muñoz-Avila, H.: Conversational case-based reasoning. Applied Intelligence 14(1), 9–32 (2001)
2. Aha, D.W., McSherry, D., Yang, Q.: Advances in conversational case-based reasoning. Knowledge Engingeering Review 20(3), 247–254 (2005)
3. Bergmann, R., Gil, Y.: Similarity assessment and efficient retrieval of semantic workflows. Inf. Syst. 40, 115–127 (2014)
4. Kohlmaier, A., Schmitt, S., Bergmann, R.: A similarity-based approach to attribute selection in user-adaptive sales dialogs. In: Aha, D.W., Watson, I. (eds.) Case-Based Reasoning Research and Development, ICCBR 2001. LNCS, vol. 2080, pp. 306–320. Springer (2001)
5. Minor, M., Montani, S., Recio-Garca, J.A.: Process-oriented case-based reasoning. Information Systems 40, 103 – 105 (2014)
6. Müller, G., Bergmann, R.: CookingCAKE: A framework for the adaptation of cooking recipes represented as workflows. In: Kendall-Morwick, J. (ed.) Workshop Proceedings from (ICCBR 2015). CEUR, vol. 1520, pp. 221–232. CEUR-WS.org (2015)
7. Müller, G., Bergmann, R.: Generalization of workflows in process-oriented case-based reasoning. In: Russell, I., Eberle, W. (eds.) Proceedings of the 28th Int. Florida Artificial Intelligence Research Society Conference, FLAIRS 2015. pp. 391–396. AAAI Press (2015)
8. Müller, G., Bergmann, R.: POQL: A new query language for process-oriented case-based reasoning. In: Bergmann, R., Görg, S., Müller, G. (eds.) Proceedings of the LWA 2015. CEUR Workshop Proceedings, vol. 1458, pp. 247–255. CEUR-WS.org (2015)
9. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of procedural knowledge from the web. In: Workshop Proceedings: WWW'12. Lyon, France (2012)
10. Zeyen, C., Müller, G., Bergmann, R.: Conversational process-oriented case-based reasoning. In: Proceedings of ICCBR 2017. LNCS, Springer (2017)

A Proposed General Formula to Create and Analyze Baking Recipes

Michael Ohene

EasierBaking,
1301 St. John Street, Apt. 206, Lafayette, LA 70506
michael@easierbaking.com
http://easierbaking.com

Abstract. A mathematical formula for characterizing baking recipes is presented as part of the 2017 Computer Cooking Contest Open Challenge. The formula produces three characteristic values, which along with common knowledge rules and classification, form the basis of two computer applications: Random Recipe Generator, which creates recipes, and Recipe Report Card, which analyzes recipes.

Keywords: recipe analysis, recipe creation, recipe classification

1 Introduction

The mystery of baking recipes has existed for many years despite many attempts to discover a formula or set of rules to describe them [1], [2]. The discovery of a universal formula or set of rules would, at least, form a basis for answering key questions governing baking. Of particular interests are the abilities to create custom recipes and to discover new uses for ingredients in baking. In lieu of a universal formula, creating new recipes by adaptation remains popular, however, this approach results in recipes limited by their reference recipe.

Adaptation has been formalized in research communities, where it involves creating new recipes by the introduction of substitute ingredients [6], primarily in a like-for-like relationship, and adaptation rules. The methods for substituting ingredients have involved evaluating the validity of substitutions by a scoring procedure [3] and by ingredient generalization through a cooking ontology [7].

This paper outlines an extended, generalized substitution process where any ingredient is a candidate for substitution. The only restrictions are common knowledge rules placed on baked good recipes (e.g., "Cobbler must not contain water", "Pie crust must contain water"). To avoid the tedious work alluded to in [7], the scope of this procedure shall be limited to baked goods.

1.1 What is a Baking Recipe?

A baking recipe provides a list of ingredients and measurements, which includes instructions for combining the ingredients. Each ingredient may be considered either a wet ingredient, a dry ingredient, or semi-wet ingredient. In the following procedure, first detailed in [5], wet and semi-wet ingredients are given constant values (see Table 1), while flavorings, leavenings (e.g., baking powder, baking soda, yeast, etc.), seasonings (e.g., salt), and food pieces (e.g., shredded coconut, walnut pieces, sesame seeds, etc.) are ignored. The constant values are multiplied by their respective measurements (usually in cups) to yield a numerical product. The products are summed and finally divided by the dry ingredient product(s), obtained from values in Table 1, to yield solutions called the moistness, fat, and egg value [5]. These characteristic values (i.e., the moistness value, the fat value, and the egg value) complete the characterization of baked good recipes.

Ingredients	**Value per Cup**
Wet Ingredients	
Water/Juice/Water/Milk	1
Butter/Oil	0.50
Banana	0.375
*Large egg (50 grams)	0.167, 1
Honey/Molasses	0.70
Dry Ingredients	
Flour (all-purpose, cocoa powder, whole-wheat)	1
Old-fashioned rolled oats	0.50
Semi-wet Ingredients	
Ground nuts (almond, pecans, walnuts)	0.33

Table 1. Constants for common wet, dry, and semi-wet ingredients. The large egg constant does not use a per cup value. *Large eggs each have a value of 0.167 in the moistness calculation and 1 in the egg calculation. Constants for common dry ingredients.

Ingredients	**Measure**	**Wet Value**	**Dry Value**
All-purpose flour, Cups(g)	2½, (352g)	—	2.50
Butter, Tbsp(g)	16, (224g)	0.50	—
Egg, #(g)	1, (50g)	0.167	—
Confectioner's sugar, Cups(g)	1½, (120g)	—	—
Vanilla extract, tsp(g)	1, (4g)	—	—
Almond extract, tsp(g)	½, (2g)	—	—
Salt, tsp(g)	½, (3g)	—	—
Baking soda, tsp(g)	1, (5g)	—	—
Cream of Tartar, tsp(g)	1, (5g)	—	—

Table 2. Mary's Sugar Cookie recipe with moistness values. [12]

$$\frac{16Tbsp * \frac{1Cup}{16Tbsp} * \frac{1}{2} + 1 * \frac{1}{6}}{2.50 * 1} = 0.27 \tag{1}$$

Equation (1) shows the wet-over-dry ingredient equation used to calculate the moistness value from *Mary's Sugar Cookie* recipe in Table 2. A similar equation is used to calculate the fat value in equation (2), only using ingredients that are considered fats. The egg value requires a number-of-eggs-per-cup-of-dry-ingredients calculation shown in equation (3).

$$\frac{16Tbsp * \frac{1Cup}{16Tbsp} * \frac{1}{2}}{2.50 * 1} = 0.20 \tag{2}$$

$$\frac{1}{2.50 * 1} = 0.4 \tag{3}$$

The general linear equation

$$(\frac{1}{q_{n+1}i_{n+1}})(q_1 i_1 + q_2 i_2 + ...q_n i_n) = [\underline{y}, \overline{y}] \tag{4}$$

defines baked goods through the use of characteristic values, where i is the ingredient constant, q is the quantity, and n is the nth ingredient. The term $[\underline{y}, \overline{y}]$ refers to the numerical range in moistness, fat, or egg value of a baked good. $\underline{y}$ represents the lower limit and $\overline{y}$ represents the upper limit of the numerical range.

1.2 Knowledge Acquisition

To accurately define the numerical ranges corresponding to baked goods, the acceptability of recipes and recipe reviews were considered. Instead of analyzing the reliability of users as in [3], the sheer number of reviews and the selection of recipe-focused review sites - as opposed to blogger-focused review sites - served to minimize unreliable reviews. The recipe review ratings and the "make it again" ratings served to define "acceptability". From this point the acceptable linear equations were constructed from equation (4) to determine the unknown constants.

From the collection of recipes, acceptable recipes tended fall within the predefined numerical ranges, thereby satisfying equation (4). Unacceptable recipes tended to fall outside the predefined numerical ranges of the baked goods. Example deviations from these generalized numerical ranges for cakes are presented in bold text in Table 3. By *generalized*, it is meant that the numerical range used for cakes in Table 3 are aggregations of several independent numerical ranges representing a variety of cakes (e.g., the egg value for pound cake only occupies a portion of the 1.00-3.50 egg range).

Recipes	Characteristic Values			Comments, (Exceptions)
	Moistness	Fat	Egg	
Cakes	0.68-1.15	0.13-0.34	1.00-3.50	
Basic 1-2-3-4 Layer	0.72	0.17	1.3	
Devil's Food	0.83	0.18	1.1	
Glazed Lemon-Thyme	**1.86**	**0.67**	2.7	
Glazed Lemon-Thyme (corrected)	0.8	0.27	1	
Confetti	**1.29**	**0.38**	2.5	Possible bad recipe
Pineapple Curry	0.7	0.13	1	
Classic Pound	0.87	0.29	2.3	
Blueberry Cornmeal	1.05	0.16	1.3	
German Chocolate	**0.64**	**0.07**	**0.2**	Possible bad recipe
Nejla's Yogurt	**0.6**	**0**	1.5	(Sponge cake)
Italian Cream	0.75	0.21	1.6	
Champagne	**0.62**	0.18	**0.5**	(Wedding cake)
Tasted Just Like Wedding	**0.59**	0.17	**0.3**	(Wedding cake)
Angel Bean Food	**0.27**	**0**	1.6	(Angel food cake)
Old-Fashioned Coconut	0.81	0.17	1.3	
Peanut Butter and Chocolate Swirl	0.88	0.25	1.5	
Pecan Crumble	0.89	0.13	1	
Guinness Stout	1.00	**0.1**	**0.9**	Possible bad recipe
Vanilla Bean Angel Food	0.97	**0**	2	(Angel food cake)
Strawberry and Cream	0.81	0.25	1.3	
Caramel	0.72	0.17	1.3	
Meyer Lemon	0.98	0.21	1.1	
Old-Fashioned Red Velvet	1.01	0.2	1.6	
Spiced Crumb	0.96	0.17	1.3	
Blood Orange	**1.3**	0.17	2	
Blood Orange (corrected)	1.13	0.17	2	
Hummingbird	0.77	0.17	1	
Strawberry Buttermilk	1.01	0.13	1.3	
Tres Leches	1.06	0.2	1.3	
Rum-Soaked	0.86	0.22	3	
Upside Down Chocolate	0.97	0.2	1.5	
Cardamom Flourless	**1.60**	**0.6**	6	Possible bad recipe
Carrot	0.91	0.17	1.3	
Pear Almond	**0.52**	0.25	1.5	
Pear Almond (corrected)	**0.67**	0.25	1.5	Possible bad recipe

Table 3. Characteristic values from the 2016 *bakeFromScratch* Special Edition. The bold values are the values that fall outside the numerical range for (*cakes*). Some recipes, labeled (*corrected*), were corrected in the online edition of the magazine after receiving reader feedback. (*Moistness*) corresponds to the thinness of the batter. [10]

2 Random Recipe Generator

Random Recipe Generator uses characteristic values to provide users with unique, randomly generated recipes. The program simply converts characteristics values to recipes.

The Random Recipe Generator functions using a clickable photo grid of baked goods and two pull-down menus. The two pull-down menus allow users to choose their "Fat Level" and "Sweetness", by choosing between "Low Fat", "Regular Fat", or "High Fat" and "Not too Sweet", "Sweet", or "Really Sweet", respectively [4].

2.1 Choosing Characteristic Values

The steps for choosing a random recipe are as follows:
1) A click by the user selects the numerical ranges that define a baked good.
2) From the user's choice for fat level, the numerical range for fat, $\boldsymbol{x_2}$, is chosen.
3) Once the numerical range for fat, $\boldsymbol{x_2} = [\underline{x}_2, \overline{x}_2]$, is chosen, a random fat value, x_2, is chosen and the other two values, the moistness value, x_1, and the egg value, x_3, are chosen according to the value x_2. Specifically,

a second value, x_3, in the numerical range for eggs, $\boldsymbol{x_3} = [\underline{x}_3, \overline{x}_3]$, is randomly chosen, which in turn automatically sets the third value, x_1
or
a second value, x_3, in the numerical range for eggs, $\boldsymbol{x_3} = [\underline{x}_3, \overline{x}_3]$, is randomly chosen, then a constant value is chosen such that a third value, x_1, lies within $[\underline{x}_1, \overline{x}_1]$.

2.2 Converting the Characteristic Values into a Recipe

After the process of choosing characteristic values based on the user input occurs, a base ingredient, i.e., an initial guess, is chosen, and the remaining ingredients are then substituted into equation (4). The possible measurements for the ingredients are defined by values in the Random Recipe Generator's database. In addition to measurement limits, the database also contains predefined, ingredient combinations. When equation (4)'s variables are replaced by quantities and ingredient constants, there exists some distance/error between the original random recipe's characteristic value vector, $\boldsymbol{x}$, and the substitution attempt's (adaptation's) characteristic value vector, $\boldsymbol{s_i}$, which can be calculated as the Euclidean distance, equation (5).

$$d(\boldsymbol{x}, \boldsymbol{s_i}) = \sqrt{(x_1 - s_{i,1})^2 + (x_2 - s_{i,2})^2 + (x_3 - s_{i,3})^2}. \tag{5}$$

There are 1410 iterations, i, of the ingredient substitution process, producing the distance values $d(\boldsymbol{x}, \boldsymbol{s_1}), ..., d(\boldsymbol{x}, \boldsymbol{s_{1410}})$. The ingredient substitution attempt (adaptation) with the shortest distance, $\arg\min d(\boldsymbol{x}, \boldsymbol{s_i})$, is selected and presented to the user.

2.3 Discovering New Ingredient Uses

Case-based reasoning differs from Random Recipe Generator's procedure, but it would be erroneous to say the current procedure did not utilize a case base. In fact, Random Recipe Generator relies upon a numerical abstraction of the recipe case base mentioned in the Knowledge Acquisition section. This abstraction helps to eliminate the detailed knowledge usually required to create recipes and completely eliminates the need for recipe retrieval.

In addition, instead of a more detailed formal concept analysis (FCA) approach described in [9], the only additional information needed to create a recipe is a generalized classification structure (e.g., whether an ingredient is a nut, egg, dairy, dry ingredient, chocolate, etc.). In other words, any ingredient can be added to the Random Recipe Generator database and incorporated into recipes as long as its classification and ingredient constant are known. As an example, Table 4 shows three recipes for chocolate chip cookies using peanut butter, ground almonds, and all-purpose wheat flour.

Ingredients	**Low Fat**		
	Not Too Sweet	**Sweet**	**Really Sweet**
All-purpose flour, Cups(g)	1 3/4, (247g)	1 1/2, (211g)	2 1/4, (317g)
Ground almonds, Cups(g)	1 3/4, (210g)	—	—
Peanut butter, Cups(g)	—	—	1/2, (129g)
Butter, Tbsp(g)	14, (196g)	8, (112g)	10, (140g)
Egg, #(g)	2, (100g)	1, (50g)	2, (100g)
Egg yolk, #(g)	—	—	—
Brown sugar, Cups(g)	2/3 Tbsp, (147g)	1/2, (110g)	1, (220g)
White sugar, Cups(g)	2/3, (133g)	1/2, (100g)	1, (200g)
Chocolate chips, Cups(g)	1 3/4, (319g)	1, (182g)	2, (365g)
Vanilla extract, tsp(g)	1 3/4, (8g)	1 1/4, (5g)	1 3/4, (8g)
Salt, tsp(g)	1/2, (3g)	1/4,(2g)	1/2, (3g)
Baking soda, tsp(g)	3/4, (4g)	1/2, (2g)	3/4, (4g)

Table 4. Three chocolate chip cookie recipes. (*Not Too Sweet*) , (*Sweet*), and (*Really Sweet*) correspond to low, normal, and high sweetness.

3 Recipe Report Card

A logical extension of the work in Table 4 is the development of a recipe analysis tool. In this role, Recipe Report Card serves to create an alternative to the traditional recipe review, i.e., to provide accurate, objective feedback for baking recipes. The use of the Recipe Report Card creates baking recipes which can be customized and prescreened. In addition, if the recipe's characteristic values fall within a predefined numerical range and satisfy common knowledge rules (e.g., "Brownie must contain chocolate"), the recipe is labeled and feedback about the recipe's sweetness and flavor is provided to the user. The predefined numerical ranges are approximated in Table 5.

4 Conclusion and Future Work

A proposed mathematical formula for baking recipes was shown capable of identifying unacceptable recipes. The results also produced logical mathematical groupings of baked good recipes. Through the Random Recipe Generator, it was shown that it is possible to generate different recipes from characteristic values via ingredient constants.

The next task for both the Recipe Report Card and the Random Recipe Generator is to produce structured lists of baking recipes. Other areas of investigation include the discovery of additional ingredient constants and the continued development of the current mathematical formula to address dairy-based desserts (e.g., ice cream, cheesecake, and custards).

References

1. Corriher, S.: Bakewise: The Hows and Whys of Successful Baking. Scribner, New York (2008)
2. Ruhlman, M.: Ratio: The Simple Codes Behind the Craft of Everyday Cooking. Scribner, New York (2010)
3. Wolf, K.I., Goetze, S., Wallhoff, F.: CooCo, What Can I Cook Today? Surprise Me. Workshop Proceedings from the Twenty-Third International Conference on Case-Based Reasoning. ICCBR 2015. Columbus (2015)
4. Ohene, M.: Michael Ohene's Cookie Recipe Generator. Cooking for Engineers, `http://www.cookingforengineers.com/article/309/Michael-Ohenes-Cookie-Recipe-Generator`
5. Ohene, M.: Analyzing a Baking Recipe. Cooking for Engineers, `http://www.cookingforengineers.com/article/280/Analyzing-a-Baking-Recipe`
6. Chef Watson, `http://www.ibmchefwatson.com`
7. Badra F., Cordier A., Lieber J.: Opportunistic Adaptation Knowledge Discovery. In: McGinty L., Wilson D.C. (eds) Case-Based Reasoning Research and Development. ICCBR 2009. LNCS, vol. 5650. Springer, Berlin, Heidelberg (2009)
8. Gaillard E., Lieber J., Nauer E.: Improving Ingredient Substitution Using Formal Concept Analysis and Adaptation of Ingredient Quantities with Mixed Linear Optimization. Proceedings from the Twenty-Third International Conference on Case-Based Reasoning. ICCBR 2015. Columbus (2015)
9. Gaillard E., Lieber J., Nauer E., Cordier A.: How Case-Based Reasoning on e-Community Knowledge Can Be Improved Thanks to Knowledge Reliability. Case-Based Reasoning Research and Development, Sep 2014, Cork, Ireland, Ireland. 8765, 155 - 169 (2014)
10. bakeFromScratch, `http://www.bakefromscratch.com/`
11. Megan's Granola, `http://allrecipes.com/recipe/98390/megans-granola/`
12. Mary's Sugar Cookies, `http://allrecipes.com/recipe/23750/marys-sugar-cookies/`

	Fat Values											
	0.00-0.05		0.05-0.10			0.10-0.20			0.20-0.34			
	Egg Values											
	0-0.5	0.5-1.0	0-0.5	0.5-1.0	1.0-2.0	0-0.5	0.5-1.0	1.0-2.0	0-0.5	0.5-1.0	1.0-2.0	2.0-3.0
Moistness												
0.00-0.05												
0.05-0.10												
0.10-0.15												
0.15-0.20						streu						
0.20-0.25			gran			cook			bcook	cook		
0.25-0.30		angel	pnuss	bisco		cook	cook		bcook	cook		
0.30-0.35	bd	chal	bd	bisco	bisco	cook	cook		bcook	cook		
0.35-0.40	bd	chal	bd	bri	bisco	pie	ging		cook			
0.40-0.45	cia	chal	sco	king	king	cobb			pie			
0.45-0.50	nokn		sco	dane				bri			brown	
0.50-0.55			biscu			cobb		bri	croiss		brown	
0.55-0.60						cobb	muff		croiss			
0.60-0.65					baba		muff	kuge				
0.65-0.70							muff	b.bd			coff	lbs
0.70-0.75					sav	eng	muff	coffee			lbs	lbs
0.75-0.80							muff	coffee				lbs
0.80-0.85								cake			tea	lbs
0.85-0.90								cake			apl	lbs
0.90-0.95								cake			ct	
0.95-1.00								cake			ct	
1.00-1.05								cake			ct	
1.05-1.10												
1.10-1.15												

Table 5. Distribution of baked good characteristic values. The following abbreviations were used. (*angel*) - angel food cake; (*apl*) - apple cake; (*baba*) - *baba al rhum*; (*b.bd*) - banana bread; (*bisco*) - biscotti; (*biscu*) - biscuit; (*bd*) - bread; (*bri*) - brioche; (*chal*) - challah; (*cia*) - ciabatta; (*ct*) - carrot cake; (*coff*) - coffee cake; (*dane*) - danish; (*cobb*) - cobbler; (*ging*) - gingerbread; (*gran*) - granola; (*king*) - king cake; (*kuge*) - *kugelhof*; (*muff*) - muffin; (*nokn*) - no knead bread; (*eng*) - Old English cake; (*pie*) - pie crust; (*pnuss*) - *pfeffernuesse*; (*lbs*) - pound cake; (*sav*) -savarin; (*sco*) - scone; (*streu*) - streusel; (*tea*) - tea cake; (*bcook*) signifies *pate brisee*, butter cookies, Mexican wedding cookies, *k'ak*, *nuhood al-adhraa*, and other eggless cookies. (*cook*) signifies chocolate chip cookies, oatmeal cookies, snickerdoodles, and other cookies that contain eggs.

Adaptation of TAAABLE to the CCC'2017 Mixology and Salad Challenges, adaptation of the cocktail names

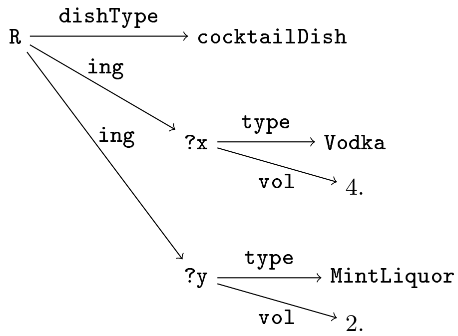
Emmanuelle Gaillard, Jean Lieber, and Emmanuel Nauer

UL, CNRS, Inria, Loria, F-54000 Nancy, firstname.lastname@loria.fr

Abstract. This paper presents the submission of the TAAABLE team to the 2017 Computer Cooking Contest. All challenges except the sandwich challenge are addressed. Online systems have been developed for the *salad* and *mixology* challenges by adapting previous successful CCC TAAABLE systems to the requirements of the 2017 challenges. However, this paper presents two main contributions. The first contribution is a new approach based on adaptation rules for managing the ingredients available for adapting salad recipes, and the second contribution is a work on cocktail name adaptation which is submitted to the *open challenge*. The cocktail name adaptation takes into account the name of the recipe which has to be adapted, knowledge about the name of the recipe and about the substituted and substituting ingredients. The naming process uses the problem-solution dependency to the context of the target problem, exploiting knowledge encoded in RDFS, a semantic Web representation language. This approach in the framework of the representation language RDFS is a general approach and can be applied in other problem solving contexts.
Keywords: case-based reasoning, adaptation, cooking, RDFS.

1 Introduction

This paper presents the participation of the TAAABLE team to the 2017 Computer Cooking Contest. TAAABLE addresses all the challenges except the sandwich challenge. Online systems have been developed for the *salad* and *mixology* challenges, respectively available on `http://tuuurbine.loria.fr/taaableCCC2017/salad.php` and `http://tuuurbine.loria.fr/taaableCCC2017/cocktail.php`. For these two challenges, previous successful CCC TAAABLE systems have been adapted to fulfill the requirements of the 2017 challenges. The management of a limited set of ingredients (called *the fridge* in the following) uses, for the mixology challenge, the process that has been presented at the CCC 2015 contest. Once the basic retrieval process of TAAABLE returns a recipe to adapt, this process based on formal concept analysis searches the more accurate available food which has to be used to substitute for a non available one. A first contribution of this paper concerns the salad challenge, for which we propose a new approach to manage the fridge, using adaptation rules. The second main contribution of this year concerns the *open challenge* for which we present a work about cocktail name adaptation. The cocktail name adaptation takes into account the



The "Green Russian" recipe is identified by the resource `R`. Its ingredients are 4 cl of vodka and 2 cl of mint liquor. The preparation is not represented. `ing` relates a recipe to one of its ingredients. `type` (abbreviation of `rdf:type`) is an RDF property relating a class to its instance (for example, the triple ⟨`?x type vodka`⟩ means that `?x` is an instance of `vodka`. The variables are existentially quantified (there exist `?x` and `?y` such that. . .). The property `vol` relates an ingredient to its volume in centilitres.

Fig. 1. An RDF graph representing the "Green Russian" cocktail recipe.

name of the recipe which has been adapted, knowledge about the name of the recipe and about the substituted and substituting ingredients. The naming process uses the problem-solution dependency to the context of the target problem, exploiting knowledge encoded in RDFS, a semantic Web representation language. This approach in the framework of the representation language RDFS is a general approach and can be applied in other problem solving contexts.

The paper is organized as follows: Section 2 introduces the core of the TAAABLE system. Sections 3, 4 and 5 present the TAAABLE systems which adress respectively the mixology, the salad and the open challenges.

2 The TAAABLE system

The challenges, proposed by the CCC since its first edition consists in proposing, according to a set of initial recipes, one or more recipes matching a user query composed of a set of wanted ingredients and a set of unwanted ingredients. Since 2015, the TAAABLE systems are built using TUUURBINE [1] (`http://tuuurbine.loria.fr`), a generic case-based reasoning (CBR) system over RDFS which allows reasoning with knowledge stored in a RDF store, as the one provided by the contest. TUUURBINE implements a generic CBR mechanism in which adaptation consists in retrieving similar cases and in replacing some features of these cases in order to adapt them as a solution to a query.

2.1 RDFS

RDF (*Resource Description Framework*) represents data as triples of resources ⟨*subject predicate object*⟩, where the resource *predicate* is a property. A resource is either a constant or a variable (generally called identified resource and blank node, respectively). By naming convention, variables start with the symbol `?` whereas constants do not. Fig. 1 illustrates a recipe represented using RDF triples.

RDFS gives some semantics—and thus, inference possibilities—to RDF by the mean of inference rules associated to some resources. Only a few rules are used in this paper:

$$\frac{\langle \mathtt{a\ type\ C}\rangle \quad \langle \mathtt{C\ subc\ D}\rangle}{\langle \mathtt{a\ type\ D}\rangle} r_1 \quad \frac{\langle \mathtt{a\ p\ b}\rangle \quad \langle \mathtt{p\ subp\ q}\rangle}{\langle \mathtt{a\ q\ b}\rangle} r_2$$

$$\frac{\langle \mathtt{A\ subc\ B}\rangle \quad \langle \mathtt{B\ subc\ C}\rangle}{\langle \mathtt{A\ subc\ C}\rangle} r_3 \quad \frac{\langle p\ \mathtt{subc}\ q\rangle \quad \langle q\ \mathtt{subc}\ r\rangle}{\langle p\ \mathtt{subc}\ r\rangle} r_4$$

`type`, `subc` and `subp` are abbreviations for `rdf:type`, `rdfs:subClassOf` and `rdfs:subPropertyOf`. `type` is the membership relation between an instance and a class. `subc` (resp., `subp`) is the relation between a class and a superclass (resp., a property and a superproperty). r_1 means that if `a` is an instance of a class it is also an instance of its superclasses. r_2 means that if `a` and `b` are related by a property, they are also related by any of its superproperties. r_3 and r_4 state that `subc` and `subp` are transitive. For example, the following inference can be drawn:

$$\left\{ \begin{array}{c} \langle \mathtt{?x\ type\ Vodka}\rangle, \\ \langle \mathtt{vodka\ subc\ Alcohol}\rangle \end{array} \right\} \vdash \langle \mathtt{?x\ type\ Alcohol}\rangle$$

RDFS does not include negation, thus only positive facts can be entailed. However, an inference with closed world assumption (CWA) can be drawn, stating that if $\mathcal{B} \not\vdash t$ then t is considered to be false (given the RDFS base $\mathcal{B}$), denoted by $\mathcal{B} \vdash_{\text{cwa}} \neg t$.

SPARQL (*SPARQL Protocol and RDF Query Language*) enables to write queries to RDF or RDFS bases. If a SPARQL engine uses RDFS entailment, this means that the query is done on the RDF base completed by RDFS entailment. For example, the following SPARQL query addressed to a base describing recipes such as the one of figure 1 returns the set of recipes `?r` containing some alcohol, taking into account the domain knowledge, in particular the subclass relations of the food hierarchy presented in Fig. 2. The CWA is assumed: if it cannot be entailed that a recipe contains some alcohol, then it is concluded that it does not.

$$\mathtt{Q}_{\mathtt{alcohol}} = \text{SELECT } \mathtt{?r} \text{ WHERE } \{\mathtt{?r\ ing\ ?a\ .\ ?a\ type\ Alcohol}\} \qquad (1)$$

Given a SPARQL query `Q` and an RDFS base $\mathcal{B}$, the result of the execution of `Q` on $\mathcal{B}$ is denoted by $\mathtt{exec}_{\vdash}(\mathtt{Q}, \mathcal{B})$.

2.2 TUUURBINE founding principles

TUUURBINE is a generic CBR system over RDFS .

The **domain knowledge** is represented by an RDFS base `DK` consisting of a set of triples of the form $\langle C\ \mathtt{subc}\ D\rangle$ where C and D are classes which belong to a same hierarchy (e.g, the food hierachy). Fig. 2 represents the domain knowledge for the running examples by a hierarchy whose edges $C \xrightarrow{x} D$ represent the triples $\langle C\ \mathtt{subc}\ D\rangle$ with x, the retrieval knowledge encoded by a cost function $\mathtt{cost}(\langle C\ \mathtt{subc}\ D\rangle) = x$. This cost can be understood intuitively as the measure of "the generalization effort" from C to D. How this cost is computed is detailed in [2].

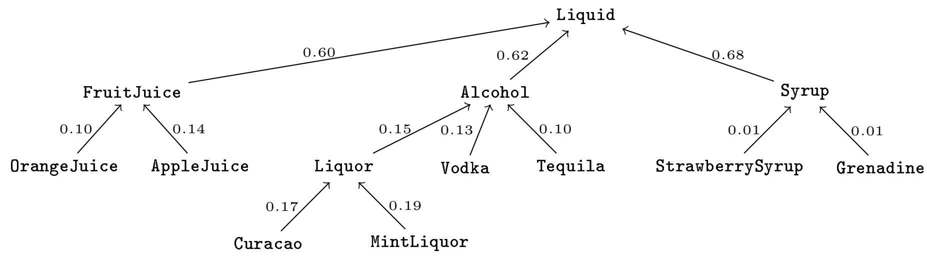


Fig. 2. The hierarchy forming the domain knowledge used in the running example with the generalization costs used as retrieval knowledge.

A **TUUURBINE case** `case` is described by a set of triples of the form $\langle URI_{\texttt{case}}\ \texttt{prop}\ \texttt{val}\rangle$, where $URI_{\texttt{case}}$ is the URI of `case`, `val` is either a resource representing a class of the ontology or a value and `prop` is an RDF property linking `case` to a hierarchy class or to the value.

For simplification, in this paper, we represent a case by a conjunction of expressions only of the form `prop : val`. For example, the "Green Russian" recipe is represented by the following index R, which means that "Green Russian" is a cocktail recipe made from vodka and mint liquor (`ing` stands for *ingredient*).

$$\texttt{R} = \texttt{dishType}:\texttt{CocktailDish} \wedge \texttt{ing}:\texttt{Vodka} \wedge \texttt{ing}:\texttt{MintLiquor} \qquad (2)$$

For instance, the first conjunct of this expression means that the triple $\langle URI_R\ \texttt{dishType}\ \texttt{CocktailDish}\rangle$ belongs to the knowledge base.

2.3 TUUURBINE query

A TUUURBINE query is a conjunction of expressions of the form `sign prop:val` where $\texttt{sign} \in \{\epsilon, +, !, -\}$, `val` is a resource representing a class of the ontology and `prop` is an RDF property belonging to the set of properties used to represent cases. For example,

$$\texttt{Q} = +\texttt{dishType}:\texttt{CocktailDish} \wedge \texttt{ing}:\texttt{Vodka} \wedge !\texttt{ing}:\texttt{Grenadine} \qquad (3)$$

is a query to search "a cocktail with vodka but without grenadine".

The signs ϵ (*empty sign*) and $+$ are "positive signs": they prefix features that the requested case must have. $+$ indicates that this feature must also occur in the source case whereas ϵ indicates that the source case may not have this feature, thus the adaptation step has to make it appear in the final case. For example, the term $+$`dishTypeCocktailDish` means that TUUURBINE will only retrieve cases which are cocktail recipes.

The signs ! and $-$ are "negative signs": they prefix features that the requested case must not have. $-$ indicates that this feature must not occur in the source case whereas ! indicates that the source case may have this feature, and, if so, that the adaptation step has to remove it.

2.4 TUUURBINE retrieval process

The retrieval process consists in searching for cases that best match the query. If an exact match exists, the corresponding cases are returned. For the query $\mathtt{Q}$ given in (3), the "Green Russian" recipe is retrieved without adaptation. Otherwise, the query is relaxed using a generalization function composed of one-step generalizations, which transforms $\mathtt{Q}$ (with a minimal cost) until at least one recipe of the case base matches $\Gamma(Q)$. A one step-generalization is denoted by $\gamma = \mathtt{prop} : \mathtt{A} \rightsquigarrow \mathtt{prop} : \mathtt{B}$, where $\mathtt{A}$ and $\mathtt{B}$ are classes belonging to the same hierarchy with $\mathtt{A} \sqsubseteq \mathtt{B}$, and $\mathtt{prop}$ is a property used in the case definition. This one step-generalization can be applied only if $\mathtt{A}$ is prefixed by ϵ or ! in $\mathtt{Q}$. If $\mathtt{A}$ is prefixed by !, thus $\mathtt{B}$ is necessarily the top class of the hierarchy. For example, the generalization of $!\mathtt{ing} : \mathtt{Grenadine}$ is $\epsilon\mathtt{ing} : \mathtt{Food}$, meaning that if grenadine is not wanted, it has to be replaced by some other food or to be removed. Classes of the query prefixed by $+$ and $-$ cannot be generalized.

Each one-step generalization is associated with a cost denoted by $\mathtt{cost}(A \rightsquigarrow B)$. The generalization Γ of $\mathtt{Q}$ is a composition of one-step generalizations $\gamma_1, \ldots \gamma_n$: $\Gamma = \gamma_n \circ \ldots \circ \gamma_1$, with $\mathtt{cost}(\Gamma) = \sum_{i=1}^{n} \mathtt{cost}(\gamma_i)$. For example, for:

$$\begin{aligned} \mathtt{Q} = &+\mathtt{dishType} : \mathtt{CocktailDish} \\ &\wedge \mathtt{ing} : \mathtt{Vodka} \wedge \mathtt{ing} : \mathtt{Curacao} \wedge !\mathtt{ing} : \mathtt{Grenadine} \end{aligned} \tag{4}$$

$\mathtt{Curacao}$ is relaxed to $\mathtt{Liquor}$ according to the domain knowledge of Fig. 2. At this first step of generalization, $\Gamma(\mathtt{Q}) = \mathtt{dishType} : \mathtt{CocktailDish} \wedge \mathtt{ing} : \mathtt{Vodka} \wedge \mathtt{ing} : \mathtt{Liquor} \wedge !\mathtt{ing} : \mathtt{Grenadine}$, which matches the recipe described in (1), indexed by $\mathtt{MintLiquor}$, which is a $\mathtt{Liquor}$.

2.5 TUUURBINE adaptation process

When the initial query does not match existing cases, the cases retrieved after generalization have to be adapted. The adaptation consists of a specialization of the generalized query produced by the retrieval step. According to $\Gamma(\mathtt{Q})$, to $\mathtt{R}$, and to $\mathtt{DK}$, the ingredient $\mathtt{MintLiquor}$ is replaced by the ingredient $\mathtt{Curacao}$ in $\mathtt{R}$ because $\mathtt{Liquor}$ of $\Gamma(\mathtt{Q})$ subsumes both $\mathtt{MintLiquor}$ and $\mathtt{Curacao}$. So, the adaptation consists in replacing curacao by mint liquor.

TUUURBINE implements also an adaptation based on rules where a rule states that in a given context $\mathcal{C}$, some ingredients $\mathcal{F}$ can be replaced by other ingredients $\mathcal{B}$. $\mathcal{C}$, $\mathcal{F}$ and $\mathcal{B}$ are the contexts, the "from part" (premise) and the "by part" (conclusion) of the adaptation rule [3]. For example, the piece of knowledge stating that, in cocktail recipes, orange juice and strawberry syrup can be replaced with pineapple juice and grenadine, can be represented by an adaptation rule with $\mathcal{C} = \mathtt{CocktailDish}$, $\mathcal{F} = \mathtt{OrangeJuice} \wedge \mathtt{StrawberrySyrup}$ and $\mathcal{B} = \mathtt{PineappleJuice} \wedge \mathtt{Grenadine}$. Such an adaptation rule can be encoded by a substitution $\sigma = \mathcal{C} \wedge \mathcal{F} \rightsquigarrow \mathcal{C} \wedge \mathcal{B}$. In the example: $\mathtt{CocktailDish} \wedge \mathtt{OrangeJuice} \wedge \mathtt{StrawberrySyrup} \rightsquigarrow \mathtt{CocktailDish} \wedge \mathtt{PineappleJuice} \wedge \mathtt{Grenadine}$. This rule-based adaptation is directly integrated in the retrieval process by searching cases indexed by the substituted ingredients for a query about the replacing ingredients, for example by searching recipes containing $\mathtt{OrangeJuice}$ and $\mathtt{StrawberrySyrup}$ for a query about $\mathtt{PineappleJuice}$ and $\mathtt{Grenadine}$.

2.6 TAAABLE as a TUUURBINE instantiation

The TAAABLE knowledge base is WIKITAAABLE (`http://wikitaaable.loria.fr/`); WIKITAAABLE is composed of the four classical knowledge containers: (1) the domain knowledge contains an ontology of the cooking domain which includes several hierarchies (about food, dish types, etc.), (2) the case base contains recipes described by their titles, the dish type they produce, the ingredients that are required, the preparation steps, etc., (3) the adaptation knowledge takes the form of adaptation rules as introduced previously, and (4) the retrieval knowledge, which is stored as cost values on subclass-of relations and adaptation rules.

In WIKITAAABLE, all the knowledge (cases, domain knowledge, costs, adaptation rules) is stored into a triple store. So, plugging TUUURBINE over the WIKITAAABLE requires only configuring TUUURBINE by giving the case base root URI, the ontology root URI and the set of properties on which reasoning may be applied.

3 Mixology challenge

The mixology challenge consists in retrieving a cocktail that matches a user query according to a set of available foods given by the CCC organizers. For this challenge, the successful TAAABLE system which won the jury and public prizes in 2015 has been adapted to take into account the new list of available foods (vodka, gin, rum, tequila, sake, champagne, tomato juice, apple juice, sparkling water, grenadine syrup, lemon juice, lime, mint, ice cube, brown sugar, salt, pepper). The principles remain the same as the ones used in the 2015 system. TUUURBINE is used to perform the retrieval step which takes into account the available foods (section 3.1). A specific adaptation step based on formal concepts is used when some ingredients of the source recipe are not available, to search the best way to replace them, or in some cases, to remove them (see Section 3.2).

3.1 Managing a fridge with TUUURBINE

TUUURBINE is able to manage a fridge directly through a query modification using the ϵ and ! prefixes. Indeed, if answers must only contain the available foods, the initial user query can be modified by adding the minimal set of classes of the food hierarchy that subsume the set of foods which are not available, each class being negatively prefixed by !. For example, let us assume that `OrangeJuice` and `AppleJuice` are the available fruit juices, that `Vodka` and `Tequila` are the only available alcohols, that `Grenadine` is the only available syrup, and that the user wants a cocktail recipe with `Vodka` but without `Grenadine`. The initial user query will be $\texttt{Q} = +\texttt{dishType}:\texttt{CocktailDish} \wedge \epsilon\texttt{ing}:\texttt{Vodka} \wedge !\texttt{ing}:\texttt{Grenadine}$. According to Fig. 2, `Liquor` and `StrawberrySyrup` will be added to this initial query with a ! for expressing that the result cannot contain one of these non available classes of food. The extended query `EQ` submitted to TUUURBINE will be:

$$\texttt{EQ} = \texttt{Q} \wedge !\texttt{ing}:\texttt{Liquor} \wedge !\texttt{ing}:\texttt{StrawberrySyrup}$$

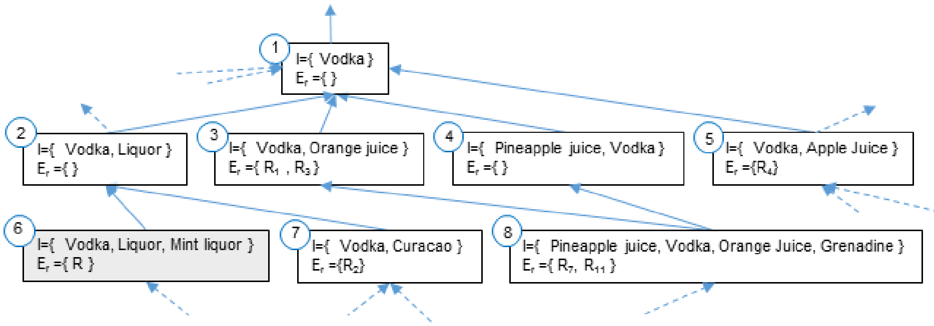


Fig. 3. Part of the concept lattice built from recipes using `Vodka`.

For this example, TUUURBINE retrieves the "Green Russian" recipe with the adaptation "replace `MintLiquor` with `Food`". In order to replace `MintLiquor` by something more specific than `Food`, a FCA approach which exploits ingredient combination in cocktail recipes is used.

3.2 Using FCA to search the best ingredient combination

FCA is a classification method allowing object grouping according to the properties they share [4] and so, is able to find regularities in a set of objects. When the objects are cocktail recipes and the properties are the ingredients the cocktail recipes use, FCA computes ingredient combination. FCA produces *formal concepts* as output. A formal concept is a pair (I, E) where I is a set of properties, E is a set of objects, respectively called the *intent* and the *extent* of the formal concept, such that (1) I is the set of all properties shared by the objects of E and (2) E is the set of all objects sharing properties in I. The formal concepts can be ordered by extent inclusion, also called *specialisation* between concepts, into what is called a *concept lattice*. Fig. 3 illustrates a part of the lattice resulting from recipes with `Vodka` (the only required ingredient in `Q`). On this figure, the extents E are given through a reduced form (noted E_r): the objects appear in the most specific concepts, the complete extent can be computed by the union of objects belonging to the subconcepts. For example, the concept #3 is related to cocktail recipes using vodka and orange juice. These recipes are R_1 and R_3 which do not contain other ingredient, and the recipes of the extents of concept #3 more specific concepts (e.g. concept #8) which contain additional ingredients. This lattice can be used to adapt the "Green Russian" recipe `R`, returned for query `Q`, with the substitution of `MintLiquor` with *another food* because `MintLiquor` is not in the set of available foods.

To search a replacing ingredient in a given recipe or in a recipe according to pieces of food that will be kept, the idea is to exploit the lattice which captures concept similarities and organization. Adapting a cocktail is based on the closeness between concepts. For example, when a replacing ingredient is searched for `MintLiquor` in `R` (concept #6), some similar concepts (i.e. sharing a same super-concept) can be used. In the lattice given in example, concept #6 can be generalized into concept number #2, which extent contains cocktails with vodka and liquor. The cocktail in the extent of concept #7 is similar to the one of concept #2, because they share the `Vodka` and the `Liquor` properties. When removing `MintLiquor` from the "Green Russian" recipe, a possible ingredient for substitution, given by the lattice, could be `Curacao`. However, as `Curacao` is not an available food, concept #7 cannot be used to complete the substitution.

Let C_R be the formal concept such that $E_r(C_R) = \{R\}$. A formal concept C close to C_R is searched according to the following procedure. C is such that its intent $I(C)$ does not contain the removed ingredient (`MintLiquor` in the example) and maximizes $|E_r(C)|$. First, C is searched in the ascendants of C_R, then in the descendants of the ascendants satisfying the available food constraints. The ingredient to be substituted is replaced by $I(C) \setminus I(C_R)$.

Applying this procedure, the most similar ingredient combinations which includes `Vodka` that can be used to replace `MintLiquor` are given by concepts #3, #4 and #5 and their descendants. However, concept #4 and its descendants cannot be used to produce a substitution because its intent contains `PineappleJuice` which is not an available food. Concept #5 intent contains `AppleJuice`, an available food, but concept #3 is closer to concept #6 than concept #5 is, according to the selection procedure based on the maximal number of objects of E_r. The cocktail system will suggest replacing the mint liquor with orange juice.

To implement our approach, data about ingredient combinations in cocktail recipes has been collected. For this, we queried *Yummly* (`http://www.yummly.com/`) with query composed of one ingredient (one available food from the CCC 2017 new food list). More details about the FCA based approach can be found in [5].

4 Salad challenge: a new approach for managing the fridge and for adapting quantities

The adaptation challenge requires managing a limited set of available food (like in the cocktail challenge) and adapting the ingredient quantities.

4.1 Salad ingredient adaptation with a fridge

The approach for managing a limited set of ingredients is rather different in the salad challenge context than in the cocktail one. Indeed, there are many important differences between the knowledge involved to solve a recipe adaptation for the mixology challenge and the knowledge involved to solve a recipe adaptation for the salad challenge. First, there are less salad recipes than cocktails recipes: 70 against 108. Second, the salad recipes use 266 different ingredients of whom 245 are not available in the fridge. For the cocktail recipes, only 139 ingredients (among the 156 different ingredients) are not in the fridge. Third, the minimal, maximal and average of ingredients per recipe is 4, 18 and 10 for the salad recipes and only 2, 10 and 5 for the cocktail recipes. The second and third points directly impact the number of ingredient substitutions required to adapt a recipe to fit the fridge constraint. Adapting a salad recipe requires at least 3 substitutions and in average 8 substitutions (the maximal number of substitutions is 17). For the cocktail recipes, to take into account the fridge, the maximal number of substitutions is only 7, 10 recipes require only 1 ingredient substitution and, in average, the number of substitutions is 3. Fifth, the ingredients involved in the salad recipes are more distributed over the food hierarchy: 13 top categories of the food hierarchy are concerned (`Vegetable`, `Fruit`, `Meat`, `Seafood`, `Legume`, `Diary`, `Liquid`, `Oil`, ...), whereas only 3 top categories are concerned for cocktail recipes (`Liquid`, `Fruit`, and `Flavoring`). All these facts drastically increase the adaptation effort for the salad challenge comparing to the cocktail challenge: more ingredients to substitute in order to adapt a recipe, more unavailable foods implied and a larger distribution of this foods in the food hierarchy. So, it is not appropriate to generalize an unavailable food to `Food` because too many food classes will be generalized into `Food`: 8 in average. With the approach presented for adapting cocktail recipes, it requires searching for each of these 8 ingredients which ingredients of the fridge can be used as substituting ingredients. That is why we propose a new approach using adaptation rules, in order to better control the adaptation. The idea is to define, for each available food f, which ingredients can be replaced by f. About 50 adaptation rules have been added to manage the foods available to cook a salad. For example, the four following adaptation rules:

$$\sigma_1 = (\mathtt{SaladDish} \wedge \mathtt{Diary} \rightsquigarrow \mathtt{SaladDish} \wedge \mathtt{Yogurt})$$
$$\sigma_2 = (\mathtt{SaladDish} \wedge \mathtt{CitrusFruit} \rightsquigarrow \mathtt{SaladDish} \wedge \mathtt{Orange})$$
$$\sigma_3 = (\mathtt{SaladDish} \wedge \mathtt{Egg} \rightsquigarrow \mathtt{SaladDish} \wedge \mathtt{Salmon})$$
$$\sigma_4 = (\mathtt{SaladDish} \wedge \mathtt{StoneFruit} \rightsquigarrow \mathtt{SaladDish} \wedge \mathtt{Strawberry})$$

state that, in salad recipes, dairy may be replaced by yogurt, citrus fruit may be replaced by orange, egg may be replaced by salmon, and stone fruit may be replaced by strawberry.

Let $\mathtt{F}$ be the set of available foods. The adaptation rules we defined are all of the form $\sigma = (\mathtt{SaladDish} \wedge \mathtt{A} \rightsquigarrow \mathtt{SaladDish} \wedge \mathtt{B})$, with $\mathtt{B} \in \mathtt{F}$. However, we can consider 3 types of rules relying on the relation between `A` and `B`:

- If $\mathtt{B} \sqsubseteq \mathtt{A}$ and $\mathtt{A} \not\sqsubseteq \mathtt{B}$ and $\nexists \mathtt{C} \in \mathtt{F}$ such that $\mathtt{B} \neq \mathtt{C}$ and $\mathtt{C} \sqsubseteq \mathtt{A}$, the adaptation rule allows to substitute all the foods more specific than `A` with `B`. For example, $\sigma = (\mathtt{SaladDish} \wedge \mathtt{Diary} \rightsquigarrow \mathtt{SaladDish} \wedge \mathtt{Yogurt})$ will allow to replace a dairy (e.g. `Creme`) appearing in a case base recipe by some yogurt. The constraint "$\nexists \mathtt{C} \in \mathtt{F}$ such as $\mathtt{B} \neq \mathtt{C}$ and $\mathtt{C} \sqsubseteq \mathtt{A}$" guarantees that there is no food in the fridge more specific than A other than B. For example, suppose that some `Cheese` (e.g. `Cheddar`) is available in the fridge, σ_1 will be split in more specific adaptatation rules, e.g. if `Cheese` and `Creme` are the only direct subclasses of `Diary`: $(\mathtt{SaladDish} \wedge \mathtt{Creme} \rightsquigarrow \mathtt{SaladDish} \wedge \mathtt{Yogurt})$, and $(\mathtt{SaladDish} \wedge \mathtt{Cheese} \rightsquigarrow \mathtt{SaladDish} \wedge \mathtt{Cheddar})$.
- If $\mathtt{B} \not\sqsubseteq \mathtt{A}$ and $\mathtt{A} \not\sqsubseteq \mathtt{B}$, the adaptation rule implies two foods which do not belong to a same category in the food hierarchy. This type of rule allows to take into account that two foods `A` and `B` play the same role in a salad dish and so, that they are substitutable. For example, σ_3 has been created because `Egg` and `Salmon` play the same role: they are proteins. This type of rule allows also to fix a food `B` as the closest available food of `A`. For example, σ_4 has been created because there is no stone fruit in the fridge and in this case, when `StoneFruit` appears in a source case recipe, the best way to substitute it with an available food is with `Strawberry`, `StoneFruit` and `Strawberry` being fruits.

- If $\texttt{B} = \texttt{Food}$, the rules state that there is no way to substitute `A` (e.g. `Tea`) appearing in a recipe of the case base with something available in the fridge. In this case, `A` is generalized into `Food` and an adaptation procedure is triggered to remove all the `A` which have been generalized into `Food`.

The impact of such rules in TUUURBINE is that new recipes are virtually created, because a recipe containing for example some grapefruit (which is a citrus fruit) will be retrieved as if this recipe contains some orange. So, the TUUURBINE retrieval process will be able to return recipes whose adaptation effort will be less costly because more controlled (the adaptation process remaining the same).

Let $\texttt{Q} = +\texttt{dishType}:\texttt{SaladDish} \wedge \texttt{ing}:\texttt{Salmon} \wedge \texttt{ing}:\texttt{Cucumber} \wedge !\texttt{Lemon}:$, be an example query meaning that the user wants a salad recipe with salmon and cucumber but without lemon. Consider the recipe named "Cucumber salad with hard-boiled egg and fromage blanc" which index is $idx(\texttt{R}) = \texttt{Egg} \wedge \texttt{FromageBlanc} \wedge \texttt{Cabbage} \wedge \texttt{Lemon} \wedge \texttt{Salt} \wedge \texttt{Pepper}$.

According to the fridge and to the food hierarchy, the unavailable foods are added to this initial query `Q` with a !. For example, `FromageBlanc` and `Egg` are unavailable classes of food. The extended query `EQ` submitted to TUUURBINE will be :

$$\texttt{EQ} = \texttt{Q} \wedge !\texttt{ing}:\texttt{FromageBlanc} \wedge !\texttt{ing}:\texttt{Egg} \wedge \ldots$$

For simplification, we present only the two classes of `EQ` which represent unavailable foods that will be taken into account by the adaptation rules. The first generalization which returns results is $\Gamma = \texttt{Cucumber} \rightsquigarrow \texttt{Vegetable}$ producing the generalized query: $\Gamma(\texttt{EQ}) = \texttt{dishType}:\texttt{SaladDish} \wedge \texttt{ing}:\texttt{Salmon} \wedge \texttt{ing}:\texttt{Vegetable} \wedge !\texttt{ing}:\texttt{Lemon} \wedge !\texttt{ing}:\texttt{FromageBlanc} \wedge !\texttt{ing}:\texttt{Egg}$. With the use of the three adaptation rules σ_1, σ_2 and σ_3, `R` matches $\Gamma(\texttt{EQ})$ because σ_1 transforms the `FromageBlanc`, a `Diary` of `R`, into `Yogurt`, σ_2 transforms the `Lemon`, a `CitrusFruit` of `R`, into `Orange`, σ_3 transforms the `Egg` of `R` into `Salmon`, and `Cabbage` is a `Vegetable` according to the food hierarchy. The answer returned by TUUURBINE for adapting `R` to `Q` consists in replacing `Cabbage` by `Cucumber`, `Egg` by `Salmon`, `FromageBlanc` by `Yogurt` and `Lemon` by `Orange`. The first subsitution comes from the basic retrieval/adaptation processes of TUUURBINE , while the three last ones come from the adaptation rules.

4.2 Adaptation of quantities for the salad challenge

The ontology-based substitution procedure extended by adaptation rules of TAAABLE favors the substitution of ingredients of the same type (a sauce by a sauce, a vegetable by a vegetable, etc.). So, ingredient quantities can, in most cases, be reused without adaptation. For example, 3 *cups* of `Pasta` can be replaced by 3 *cups* of `Couscous`, 1 *tsp* of `Oregano` by 1 *tsp* of `Cumin`, etc. However, there are some kinds of adaptations (coming from the adaptation rules) which require some quantity adjustements. An approach for the adaptation of ingredient quantities based on mixed linear optimization was proposed in [6] and has been used to compute sugar, alcohol and mass compensation when replacing some ingredients by others in the context of cocktail adaptations [5]. This complexity is not required to adapt salads, especially because the substitution procedure is guided by adaptation rules. So, we only define a simple heuristic to provide realistic quantities to the user. This heuristic is the following. Each food available in the fridge is associated with a preferred unit and to a set of possible units coming from the recipes of the case base. For example, the set of possible units for `Carrot` is $\{unit, g, oz\}$ and the preferred unit is $unit$. For the quantity adaptation, if the replaced ingredient unit is a mass (e.g. g) or a volume (e.g. cl, tsp, $tblsp$) in the source case, and if this unit is a possible unit for the replacing ingredient, neither quantity adaptation nor unit adaptation is done. If not, conversion knowledge coming from WIKITAAABLE is used. First, the quantity of the replaced ingredient is converted into its mass, in grams. For example, if the source recipe uses 2 lettuces, the conversion knowledge stating that the mass of 1 lettuce is 360 g is used to computed the total mass of lettuce: $2 \times 360\ g = 720\ g$). Secondly, conversion knowledge associates to each possible replacing ingredient (a food of the fridge) its preferred unit and the correspondance from this unit to a mass in grams. For example, the preferred unit of `Carrot` is $unit$ and the mass of 1 $unit$ of carrot is 70 g. This allows computing the adapted quantity in the preferred unit of the replacing ingredient, by divided the mass (in grams) of the replaced ingredient by the mass (in grams) corresponding to the preferred unit of the replacing ingredient. In the running example, if `Lettuce` is replaced by `Carrot`, and the original quantity of `Lettuce` is 2 $units$ then $720/70 \approx 10$ $units$ of `Carrot` have to be used (the result is rounded).

5 Open Challenge: cocktail name adaptation

This section presents the TAAABLE team submission to the open challenge. The issue is, when TAAABLE returns an adapted recipe, how to name it according to its original name and to the substitution. For example, what name could be assigned to the recipe obtained from the “Green Russian” recipe after having substituted the mint liquor with curacao.

This issue of adapting recipe names has been suggested by the jury of the 2014 edition of the CCC and the CookingCAKE system [7] has addressed this challenge for the CCC-2015, using a few rules. For instance, the adjective `"cheesy"` is added to the recipe name if the adapted recipe of a sandwich contains some cheese. The TAAABLE team has addressed this issue more formally in [8], using an approach based on RDFS. In this application, a problem `pb` is a representation of a cocktail recipe by an RDFS graph. For the first version of this application, only ingredient types are considered, neither the quantities, nor the preparation steps. Therefore, a problem is an RDFS base $\mathtt{pb} = \bigcup_{k=1}^{n} \{\langle id\ \ \mathtt{ing}\ \ \mathtt{?v}_k\rangle, \langle \mathtt{?v}_k\ \ \mathtt{type}\ \ f_k\rangle\}$ where id is a constant (a resource identifying the recipe), $\mathtt{?v}_1, \ldots, \mathtt{?v}_n$ are n variables, and $f_1, \ldots, f_n$ are food classes.

A solution $\mathtt{sol}(\mathtt{pb})$ of `pb` is a literal of type string that gives a name to `pb`. It is assumed to be in lower case for the sake of simplicity, e.g., $\mathtt{sol}(\mathtt{pb}) =$ `"green russian"` solves the problem `pb` represented in figure 1. The following operations on strings are used: concatenation (denoted by $+$, e.g., `"ab"` $+$ `"cd"` $=$ `"abcd"`), substring checking (denoted by `subStringOf`, e.g., `subStringOf("bc", "abcd")` $=$ `true`), and string replacement (e.g., `replace("ab", "cd", "bababa")` $=$ `"bcdcda"`).

A dependency $\beta_{\texttt{pb}}$ between $\texttt{pb}$ and $\texttt{sol}(\texttt{pb})$ is an RDFS base. Usually, at least one food class f_k of $\texttt{pb}$ and the literal $\texttt{sol}(\texttt{pb})$ occurs in $\beta_{\texttt{pb}}$: when it is not the case, $\beta_{\texttt{pb}}$ does not relate $\texttt{pb}$ to $\texttt{sol}(\texttt{pb})$ (which is possible, e.g., when $\beta_{\texttt{pb}} = \emptyset$, i.e., there is no known dependency between $\texttt{pb}$ and $\texttt{sol}(\texttt{pb})$). For each case $(\texttt{srce}, \texttt{sol}(\texttt{srce}))$, $\beta_{\texttt{srce}}$ is assumed to be given.

A matching $\alpha_{\texttt{pb}}$ from $\texttt{srce}$ to $\texttt{tgt}$ is either simple or complex. A simple matching has the form $f \rightsquigarrow g$ where f is a food class of $\texttt{srce}$ and g is a food class of $\texttt{tgt}$; it represents the substitution of f by g. The removal of a food class f will be denoted by $f \rightsquigarrow \emptyset$. A complex matching is a composition $\alpha_{\texttt{pb}} = \alpha^{q}_{\texttt{pb}} \circ \alpha^{q-1}_{\texttt{pb}} \circ \ldots \circ \alpha^{1}_{\texttt{pb}}$ of simple matchings. $\alpha_{\texttt{pb}}$ is built during the adaptation of ingredients process of TAAABLE .

The matching α_β from $\beta_{\texttt{srce}}$ to $\beta_{\texttt{tgt}}$ is built during the cocktail name adaptation. It consists of a set of ordered pairs (d, d') where d is a descriptor of $\beta_{\texttt{srce}}$ and d' is a descriptor of $\beta_{\texttt{tgt}}$, a descriptor being either a resource (that can be a property) or a literal.

We present in the next sections five adaptation strategies: the two first strategies (§5.1 and §5.2) are application-dependent, whereas the last ones should be adaptable to other applications. Strategies presented in sections 5.3 and 5.4 are designed for simple matchings whereas the strategy of section 5.5 combines strategies for dealing with complex matchings.

5.1 Strategy "Alcohol abuse is dangerous for health"

Consider a cocktail recipe containing some alcohol, for which the adaptation consists in removing the ingredients which are alcohol or in substituting them by ingredients which are not alcohol. In this case, the new cocktail name may be computed by adding "virgin" to the original recipe name. For example, let $\texttt{sol}(\texttt{srce}) = \texttt{"mojito"}$ be the name of the "Mojito" recipe, let $\alpha_{\texttt{pb}} = \texttt{rhum} \rightsquigarrow \emptyset$ be the adaptation consisting in removing the unique alcohol of $\texttt{srce}$, then $\texttt{sol}(\texttt{tgt}) = \texttt{"virgin mojito"}$ will be the adapted cocktail name. Note that the test about alcohol can be performed by executing the SPARQL query $\texttt{Q}_{\texttt{alcohol}}$ (cf. equation (1)) twice:

- "$\texttt{srce}$ contains some alcohol" is encoded by $\texttt{exec}_{\vdash}(\texttt{Q}_{\texttt{alcohol}}, \texttt{DK} \cup \texttt{srce}) \neq \emptyset$ and
- "$\texttt{tgt}$ contains no alcohol" is encoded by $\texttt{exec}_{\vdash}(\texttt{Q}_{\texttt{alcohol}}, \texttt{DK} \cup \texttt{tgt}) = \emptyset$.

5.2 Default strategy

The default strategy is applied when all other strategies fail, this time by adding "the new " to the original recipe name (e.g. "the new bloody mary").

5.3 Strategy "Turn constants into variables"

This section models the adaptation example presented in Section 2, where the "Green Russian" recipe is adapted by replacing mint liquor by curacao.

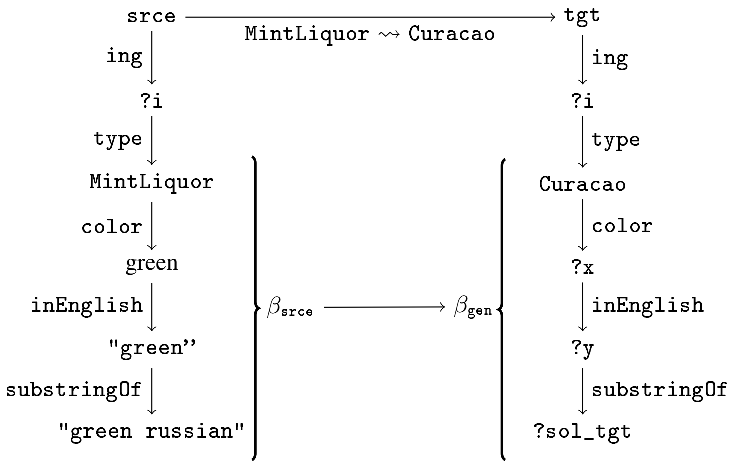


Fig. 4. Example of an RDFS based cocktail name adaptation using the "Turn constants into variables" strategy.

A partial explanation of the name $\texttt{sol}(\texttt{srce}) = $ `"green russian"` is that the color of mint liquor is green, which can be modeled by:

$$\beta_{\texttt{srce}} = \{\langle\texttt{MintLiquor color green}\rangle, \langle\texttt{green inEnglish "green"}\rangle,\\ \langle\texttt{"green" subStringOf "green russian"}\rangle\}$$

These triples are represented on the left-hand part of the graph of Fig. 4.

Since $\alpha_{\texttt{pb}} = \texttt{MintLiquor} \rightsquigarrow \texttt{Curaçao}$, in order to build $\beta_{\texttt{tgt}}$, the idea is to apply $\alpha_{\texttt{pb}}$ on $\beta_{\texttt{srce}}$ and then to make some modifications on the resources and literals to make it consistent with DK. This consistency test must be considered wrt CWA because there is no way to have $\langle\texttt{MintLiquor color blue}\rangle$ inconsistent with DK in the classical semantics. It is assumed that $\texttt{DK} \vdash_{\text{cwa}} \neg\langle\texttt{MintLiquor color blue}\rangle$, thus the mere substitution $\alpha_{\texttt{pb}}$ on $\beta_{\texttt{srce}}$ gives an inconsistent result wrt DK under CWA. So, the idea is to relax this triple. One way to do it is to replace `green` with a variable `?x`. More generally, the strategy consists in replacing the descriptors of $\beta_{\texttt{srce}}$ by variables, with the exception of the predicates (that are higher order resources) and of the descriptors occurring in `tgt`. The variable that replaces `sol(srce)` is `?solTgt`: solving `tgt` consists in giving a value `sol(tgt)` to this variable. This gives the following dependency (obtained by applying $\alpha_{\texttt{pb}}$ and turning some constants into variables):

$$\beta_{\texttt{gen}} = \{\langle\texttt{Curacao color ?x}\rangle, \langle\texttt{?x inEnglish ?y}\rangle,\\ \langle\texttt{?y subStringOf ?solTgt}\rangle\}$$

$\beta_{\texttt{gen}}$ is so-called, since it generalizes $\alpha_{\texttt{pb}}(\beta_{\texttt{srce}})$ (in the sense $\alpha_{\texttt{pb}}(\beta_{\texttt{srce}}) \vdash \beta_{\texttt{gen}}$), where $\alpha_{\texttt{pb}}(\beta_{\texttt{srce}})$ is the result of applying the substitution $\alpha_{\texttt{pb}}$ on $\beta_{\texttt{srce}}$.

Now, in order to get $\beta_{\texttt{tgt}}$, the idea is to unify the variables `?x` and `?y` with some constants, using the domain knowledge. Therefore DK is interrogated with the following SPARQL query: `SELECT ?x ?y WHERE {Curacao color ?x . ?x inEnglish ?y}`.

Assuming the only result is the pair $\{\texttt{?x} \leftarrow \texttt{blue}, \texttt{?y} \leftarrow \texttt{"blue"}\}$, it comes:

$$\beta_{\texttt{tgt}} = \{\langle\texttt{Curacao color blue}\rangle, \langle\texttt{blue inEnglish "blue"}\rangle,$$
$$\langle\texttt{"blue" subStringOf ?solTgt}\rangle\}$$
$$\text{and } \alpha_\beta = \{(\texttt{MintLiquor}, \texttt{Curacao}), (\texttt{green}, \texttt{blue}), (\texttt{"green"}, \texttt{"blue"})\}$$

Therefore, $\beta_{\texttt{tgt}}$ involves that $\texttt{sol}(\texttt{tgt})$ has to respect the following constraint:

$$\texttt{sol}(\texttt{tgt}) \in \{s : \text{string} \mid \texttt{"blue"} \text{ is a substring of } s\} \tag{5}$$

Now, $\texttt{sol}(\texttt{srce})$ must be modified using α_β into $\texttt{sol}(\texttt{tgt})$ that respects (5). Here, a domain-dependent choice is made: it concerns the way the solution space is structured, i.e., how can modifications be applied on solutions. It is assumed that in this application, the only modification operation is based on the `replace` operation on the set of strings (which is the solution space). Hence, since $(\texttt{"green"}, \texttt{"blue"}) \in \alpha_\beta$, the following cocktail name that is consistent with (5) is proposed:

$$\texttt{sol}(\texttt{tgt}) = \texttt{replace}(\texttt{"green"}, \texttt{"blue"}, \texttt{sol}(\texttt{srce})) = \texttt{"blue russian"}$$

5.4 Strategy "Generalization-specialization of dependencies"

Now, consider the example of the adaptation of $\texttt{sol}(\texttt{srce}) = \texttt{"green russian"}$ when $\alpha_\beta = \texttt{MintLiquor} \rightsquigarrow \texttt{IndianTonic}$ with the same $\beta_{\texttt{srce}}$ as in section 5.3 and assuming that DK gives no color to Indian tonic (i.e., there is no triple of the form $t = \langle\texttt{IndianTonic color c}\rangle$ such that $\texttt{DK} \vdash t$), the adaptation strategy of section 5.3 fails. However, it is assumed that

$$\texttt{DK} \vdash \left\{\begin{array}{c}\langle\texttt{IndianTonic taste bitter}\rangle, \langle\texttt{IndianTonic texture sparkling}\rangle,\\ \langle\texttt{bitter inEnglish "bitter"}\rangle, \langle\texttt{sparkling inEnglish "sparkling"}\rangle,\\ \langle\texttt{color subp hOP}\rangle, \langle\texttt{taste subp hOP}\rangle, \langle\texttt{texture subp hOP}\rangle\end{array}\right\}$$

meaning that Indian tonic is bitter and sparkling, and that color, taste and texture are organoleptic properties (`hOP` is an abbreviation for `hasOrganolepticProperty`). Therefore, the adaptation strategy described in section 5.3 can be applied with a slight modification: it is sufficient to replace in $\beta_{\texttt{gen}}$ the triple $\langle\texttt{IndianTonic color ?x}\rangle$ by $\langle\texttt{IndianTonic hOP ?x}\rangle$, which is more general according to DK.

One way to address this problem is to search in the domain knowledge for triples for building $\beta_{\texttt{gen}}$ that are *similar* to $\alpha_\beta(\beta_{\texttt{srce}})$. This can be likened to the retrieval issue in CBR, which can be implemented by a least generalization of the query (see, e.g., [9]). A similar idea is proposed here. It consists in making a best-first search in a space of dependencies β such that:

- The initial state β_0 corresponds to the $\beta_{\texttt{gen}}$ as it is computed in the strategy of section 5.3.
- The successors of a state consists in making a generalization of one of its triples. The following generalization operators can be considered: replace a class (resp., a property) by a direct superclass (resp., direct superproperty) in DK, replace a resource or a literal by a variable, etc. A cost function must be associated to generalization operators, in order to choose the least costly generalization.

– A final state β is such that the SPARQL query associated with it gives a nonempty set of results.

Once a final state β is found, the rest of the approach of Section 5.3 can be applied with $\beta_{\texttt{gen}} = \beta$.

Back to the example, it comes:

$$\beta_0 = \{\langle \texttt{IndianTonic color ?x}\rangle, \langle \texttt{?x inEnglish ?y}\rangle,$$
$$\langle \texttt{?y subStringOf ?solTgt}\rangle\}$$

In the first triple, `color` can be generalized into `hOP` (since $\texttt{DK} \vdash \langle \texttt{color subp hOP}\rangle$), giving

$$\beta = \{\langle \texttt{IndianTonic hOP ?x}\rangle, \langle \texttt{?x inEnglish ?y}\rangle,$$
$$\langle \texttt{?y subStringOf ?solTgt}\rangle\}$$

β is a final state since $\texttt{exec}_{\vdash}(\texttt{Q}, \texttt{DK}) \neq \emptyset$ for

$$\texttt{Q} = \text{SELECT } \texttt{?x ?y} \text{ WHERE } \{\texttt{IndianTonic hOP ?x . ?x inEnglish ?y}\}$$

Indeed, $\texttt{exec}_{\vdash}(\texttt{Q}, \texttt{DK}) = \{A_1, A_2\}$ where $A_1 = \{\texttt{?x} \leftarrow \texttt{bitter}, \texttt{?y} \leftarrow \texttt{"bitter"}\}$ and $A_2 = \{\texttt{?x} \leftarrow \texttt{sparkling}, \texttt{?y} \leftarrow \texttt{"sparkling"}\}$, leading to the two expected solutions: `"bitter russian"` and `"sparkling russian"`.

Therefore this strategy consists in finding the minimal generalization β of the initial dependency β_0 and then in specializing β into $\beta_{\texttt{tgt}}$'s thanks to SPARQL querying on `DK`, hence the name of the strategy.

5.5 Composing strategies when the matching is complex

When the matching $\alpha_{\texttt{pb}}$ is complex, it can be written $\alpha_{\texttt{pb}} = \alpha_{\texttt{pb}}^q \circ \alpha_{\texttt{pb}}^{q-1} \circ \ldots \circ \alpha_{\texttt{pb}}^1$, with $q \geq 2$. The idea is then to apply in sequence the strategies associated with simple matchings. For example, for $\texttt{sol}(\texttt{srce}) = \texttt{"green russian"}$, $\alpha_{\texttt{pb}}^1 = \texttt{mintLiquor} \rightsquigarrow \texttt{curacao}$, $\alpha_{\texttt{pb}}^2 = \texttt{vodka} \rightsquigarrow \texttt{tequila}$, the strategy of Section 5.3 can be applied twice to give the name $\texttt{sol}(\texttt{tgt}) = \texttt{"blue mexican"}$. This adaptation is an application of the adaptation based on reformulations and similarity paths (see e.g. [10]).

6 Conclusion

This paper has presented the systems developed by the TAAABLE team for its participation to the 2017 CCC. The two systems presented for the *salad* and *mixology* challenges are based on the successful 2015 version of TAAABLE , extended for *salad* challenge with a new approach to manage the fridge. A new approach has also been presented for adapting the cocktail names from the ingredient adaptation. Several name adaptation strategies have been presented and, if some proposed strategies are application-dependent, it is claimed that other ones can be applied—or adapted—to a larger framework. Indeed, they match the principles described in some related work about analogical transfer (e.g., [11] and [12]) while proposing an approach benefitting from the standard RDFS and associated tools (RDFS SPARQL engines, RDF stores). A first prototype implementing the three first strategies has already been developed, but the adaptation strategy based on generalization-specialization of dependencies is under development. However, there is an important workload for acquiring dependencies $\beta_{\mathtt{srce}}$, which is currently done manually and for acquiring triples in the domain knowledge. A possibility to address these issues is to query the Linked Open Data (LOD), a huge cloud of RDF and RDFS bases freely accessible on the Web. This knowledge acquisition task is the main future work.

References

1. E. Gaillard, L. Infante-Blanco, J. Lieber, and E. Nauer. Tuuurbine: A Generic CBR Engine over RDFS. In *Case-Based Reasoning Research and Development*, volume 8765, pages 140 – 154, Cork, Ireland, September 2014.
2. A. Cordier, V. Dufour-Lussier, J. Lieber, E. Nauer, F. Badra, J. Cojan, E. Gaillard, L. Infante-Blanco, P. Molli, A. Napoli, and H. Skaf-Molli. Taaable: a Case-Based System for personalized Cooking. In S. Montani and L. C. Jain, editors, *Successful Case-based Reasoning Applications-2*, volume 494 of *Studies in Computational Intelligence*, pages 121–162. Springer, 2014.
3. E. Gaillard, J. Lieber, and E. Nauer. Adaptation knowledge discovery for cooking using closed itemset extraction. In *The Eighth International Conference on Concept Lattices and their Applications - CLA 2011*, pages 87–99, 2011.
4. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin, 1999.
5. E. Gaillard, J. Lieber, and E. Nauer. Improving Ingredient Substitution using Formal Concept Analysis and Adaptation of Ingredient Quantities with Mixed Linear Optimization. In *Computer Cooking Contest Workshop*, Frankfort, Germany, September 2015.
6. J. Cojan and J. Lieber. Applying Belief Revision to Case-Based Reasoning. In *Computational Approaches to Analogical Reasoning: Current Trends*, volume 548 of *Studies in Computational Intelligence*, pages 133 – 161. Springer, 2014.
7. G. Müller and R. Bergmann. CookingCAKE: A Framework for the adaptation of cooking recipes represented as workflows. 2015.
8. N. Kiani, J. Lieber, E. Nauer, and J. Schneider. . In A. K. Goel et al., editor, *Case-Based Reasoning Research and Development, Proceedings of the 24th International Conference on Case-Based Reasoning (ICCBR-2016)*, Lecture Notes in Computer Science, Atlanta, Georgia, 2016. Springer International Publishing.
9. E. Gaillard, L. Infante-Blanco, J. Lieber, and E. Nauer. Tuuurbine: A Generic CBR Engine over RDFS. In *Case-Based Reasoning Research and Development*, volume 8765, pages 140 – 154, Cork, Ireland, September 2014.
10. E. Melis, J. Lieber, and A. Napoli. Reformulation in Case-Based Reasoning. In B. Smyth and P. Cunningham, editors, *Fourth European Workshop on Case-Based Reasoning, EWCBR-98*, LNCS 1488, pages 172–183. Springer, 1998.
11. D. Gentner. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive science*, 7(2):155–170, 1983.
12. P. H. Winston. Learning and reasoning by analogy. *Communications of the ACM*, 23(12):689–703, 1980.

Cooking On The Margins: Probabilistic Soft Logics for Recommending and Adapting Recipes

Johnathan Pagnutti and Jim Whitehead

{jpagnutt, ejw}@ucsc.edu
University of California, Santa Cruz
1156 High St
Santa Cruz, CA, 95064

Abstract. This paper introduces InclinedChef[1], for the mixology and open challenges as part of the Computer Cooking Competition. InclinedChef uses Probabilistic Soft Logics (PSLs), a logic formalism that relaxes boolean logic operations to probabilities. PSLs have had good success in recommendation engines, but have yet to be applied to the Case Based Reasoning (CBR) domain. They show a lot of promise for their ability to handle contradictory information, multiple data sources and shifting user constraints.

1 Introduction

When it comes to cooking with computers, one of the core challenges is dealing with user preferences. The decision to decide to prepare a certain meal over another one is multifaceted, with often competing preferences. The Eating Motivation Survey contains a comprehensive model of what motivates eating and food selection, comprised of 78 individual motivations collected into 15 general clusters[10]. In this study, top food selection motivations include liking a food (taste), habits, health, convenience, and pleasure. Other research confirms taste, convenience and price as being at least as important as healthiness when people pick food choices[11].

This leads to a set of hurdles when it comes to recommending and adapting recipes for users. Information about each of these motivations often is located in separate ontologies, databases, or websites. Each of these ontologies may have contradictory information, for example: a 'snack' in the What We Eat In America (WWEIA) survey is different from what a 'snack' is in the wikiTaaable ontology[1,3]. To complicate the problem even further, these preferences do not often line up: a healthy meal choice may be the least convenient option. Simple rule based formulations and single-heuristic optimizations cannot capture the fuzziness of the recipe domain.

We propose to use Probabilistic Soft Logics (PSLs)[2] to encode ontology information and a PSL solver to recommend and adapt recipes. A PSL program is a set of logical implications, written in a Prolog-like first-order logic syntax. This

[1] http://tinyai.net/cocktails

paper will introduce InclinedChef, a system that uses PSLs for recommending and adapting recipes. We will provide a short introduction into PSLs, how to encode ontology information as PSL atoms and predicates, using a PSL solver to find highly probable results for a query and highly probable adaptations for those results. We will conclude by looking at future potential applications and improvements.

2 Related Work

The use of first order logic to represent Case Based Reasoning (CBR) tasks is not novel. Delgrande added the 'default' operator to a classical first-order logic to help handle common-sense reasoning tasks[4]. CHEF[7], something of the grandaddy of CBR cooking systems, had rules that could be represented as first-order logic statements (both CHEF's rule-based simulations of cooking and its adaptation reasoning could be expressed as a implications). Nearly every CBR system in the computer cooking competition has used rule-based formalisms to some degree, with implication (a implies b) being a core component.

Other computer cooking systems have used bottom-up learning approaches, which at their core, are inferring likely ingredient substitutions from datasets of recipes. PIERRE[9] uses a neural net to learn a regression model to find highly rated combinations of ingredients. It then uses the regression model along with a genetic algorithm to come up with new recipes.

PSLs express facts, relationships and implications using first-order logic, but with the twist that all facts and relationships have an associated probability of being true, and implications have associated weights. PSL programs are internally converted into hinge-loss Markov random fields, which are evaluated as a kind of convex optimization problem. PSLs have been used to combine several evaluation metrics for recommending restaurants and songs[8], and for learning weights on implication rules from data for textual sentiment[5].

3 Probabilistic Soft Logics

Probabilistic soft logics are composed of two parts, a set of predicates that describe potential facts about the world, and a set of logical implications that relate these predicates together. PSLs are different from other logic based programming paradigms in that:

1. Logical predicates are soft. Instead of predicates returning if an atom is true or false, they return the probability that an atom being true or false.
2. PSLs require that all potential atoms be represented, even those that are impossible. Therefore, PSL formulations of problems tend to be space inefficient.
3. Going along with soft predicates, implications have weights. These weights can be learned based on ground truth, or they can be used to infer the probability of new predicates given a current set. InclinedChef uses hand-tuned implication weights to infer the probability of new predicates.

For more detail on how InclinedChef works, see System Overview (section 4).

A PSL solver converts weighted implication rules and predicates to a Hinge-Loss Markov Random Field (HL-MRF). To discuss how, we need to first look at a PSL predicate.

$$Friends(\text{“}Jon\text{”}, \text{“}Delilah\text{”}) = 0.7$$

$$Friends(\text{“}Delilah\text{”}, \text{“}Philip\text{”})$$

The first atom states that Jon and Delilah are friends (they're arguments to the Friends predicate) with a probability of 70%. The second line states that Delilah and Philip are friends, but we don't yet know how likely that predicate is. Any atoms that are given a probability before the solver runs are treated as hard constraints. The probability is held fixed while finding the probability of the remaining predicates. This allows PSLs to pull in knowledge. Using an example from InclinedChef:

$$IngGeneralizes(\text{“}Game\text{”}, \text{“}Meat\text{”}) = 0.748$$

This states that we can generalize “game" as a “meat" with fairly high confidence. This value comes from the wikiTaaaable ontology (namely, Generalization_costs). More information is in System Overview (section 4).

Because we'd like to discuss atoms without needing to write out their arguments, we'll use the convention Friends/2, which states that the Friends predicate takes two atoms as arguments. Lets look at a PSL implication rule. For example:

$$3 : Friends(X, Y) \wedge Friends(Y, Z) \implies Friends(X, Z)$$

This rule states that friendship is transitive: if Jon is friends with Delilah and Delilah is friends with Philip, it's likely that Jon is also friends with Philip. We also have a weight on this rule, which is how important it is in relation to other rules. Rules can also be hard constraints that must be upheld when inferring the probability of unknown atoms; hard constraints follow a slightly different syntax, using another example from an older version InclinedChef:

$$MustContain(I) \implies TargetIngredients(I).$$

This rule states that all MustContain/1 predicates must also be TargetIngredient/1 predicates. InclinedChef uses these constraints to insure that certain ingredients are present or not present in the adapted recipes, with more information in System Overview (section 4).

Given a set of predicates and implications, the PSL solver relaxes the boolean logic operators to continuous variables. For example, to the PSL solver, $a \implies b$ is relaxed to $max(a-b, 0)$ and $a \wedge b$ is relaxed to $max(a+b-1, 0)$. This converts the logic operators and predicates to a set of continuous valued variables, which allows the solver to define an HL-MRF over the set of unknown Y variables conditioned on the known X variables, such that the probability of Y given X is related to a hinge-loss potential function, which the solver solves using convex optimization. Due to the nature of the problem, this is parallelizable and decently fast.

4 System Overview

The system work can be divided into two parts—offline conversion and online recommendation and adaptation. Because PSL programs need to be in their own syntax, information from datasets and ontologies needs to be converted to atoms and rules. After this step, the solver can use the rules to perform recommendation and adaptation.

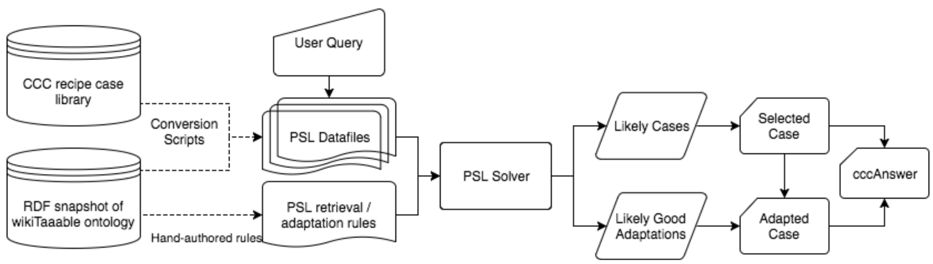


Fig. 1. Diagram of ImpliedChef. Dashed lines are offline, solid lines are online. Using conversion scripts and hand authoring, the XML case library and RDF ontology snapshot are converted to PSL atoms and rules. A user query is converted to a set of PSL atoms as well. This is passed to the solver, which returns as PSL atoms a set of likely good cases and likely good adaptations. These are used for case retrieval and case adaptation, which are then passed back as an answer to the query.

Figure 4 shows a boxes and arrows diagram of ImpliedChef. Offline (before any user queries are processed), we use a set of conversion scripts and hand authoring to get a set of PSL atoms, predicates and rules from the wikiTaaable ontology and Computer Cooking Competition case library. Online, a user query is converted to a set of PSL atoms that interact with PSL logic rules. All of this is passed to the PSL solver, which is given two important targets: a set of atoms that represent each case in the case library and a set of atoms that represent recipe adaptations. These are not the only atoms that the solver is inferring probabilities for, but are used in the next steps.

The case atoms are used to select a case from the library. The adaptation atoms are used to figure out how to tweak the case to fit the constraints provided by the user query as well as the the challenge at hand. Finally, these are wrapped up as a XML document that follows the cccSystemOutput XML Schema definition for the competition.

4.1 Offline Components

These parts of ImpliedChef are performed only once, and are done before any user queries are ever fielded by the system. The first few of these convert recipes into sets of ingredient atoms, as shown in figure 4.1.

We specify that the IngInRecipe/2 predicate is closed, which means the solver should treat all predicates of that type as observed, true data, and furthermore,

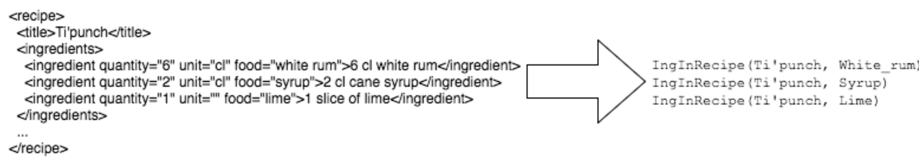


Fig. 2. Converting CCC XML case libraries into PSL atoms. We focused on ingredients for ImpliedChef

should not attempt to infer probabilities on them. IngInRecipe/2 simply encodes which recipes contain which ingredients. We discuss, under Future Work, how PSL might consider the amounts of each ingredient in its solving steps. We also use a set of scripts that converts information from the RDF snapshot of the wikiTaaable ontology to PSL atoms, for the IngGeneralizes/2 predicate. A few examples are:

$$IngGeneralizes(Pork\_side\_cuts, Pork) = 0.968$$
$$IngGeneralizes(Red\_martini, Martini) = 0.500$$
$$IngGeneralizes(Fresh\_bean, Vegetable) = 0.422$$

These atoms are calculated by subtracting the Generalization_cost of an ingredient or ingredient class by 1. These Generalization_costs are retrieved from the RDF snapshot. However, not all ingredients have Generalization_costs in the wikiTaaable ontology. For those that do not, we use the subclassOf feature and assign only a probability of 0.5 to this being a correct way to generalize about an ingredient.

The result is that, as the Generalization_cost decreases, the more the solver considers a generalization as likely. Greater generalization costs (representing ingredient classes further away from each other) are less and less likely. The RDF snapshot only provides costs for ingredients next to each other, but it can be useful for retrieval and adaptation to consider generalizations further away, such as IngGeneralizes(Rye_whiskey, Alcohol) or IngGeneralizes(Veal_leg_cut, Meat). We append these case generalizations without providing probabilities, and use the following rule to let the solver figure out how likely each of these cases should be:

$$IngGeneralizes(C1, C2) \wedge IngGeneralizes(C2, C3) \implies IngGeneralizes(C1, C3)$$

4.2 Online Components

After a user query is parsed, it is turned into two sets of atoms for the predicates MustContain/1 and MustNotContain/1. These contain the ingredients specified by the user query about ingredients they want and do not want in a resultant recipe. For retrieval, these are considered soft constraints, as we can adapt any ill-fitting recipes, but for adaptation, they are hard constraints. In addition, to fit with the cocktail challenge, we added the reduced list of ingredients as a set of MustContain/1 atoms.

For retrieval, we'd like to consider each recipe in terms of the classes that its ingredients generalize to. We capture this with a few rules:

$$\begin{aligned}
IngInRecipe(R, I) \wedge IngGeneralizes(I, C) &\implies RecipeClasses(R, C) \\
RecipeClasses(R, C1) \wedge IngGeneralizes(C1, C2) &\implies RecipeClasses(R, C2) \\
MustContain(I) \wedge IngInRecipe(R, I) &\implies RecommendTarget(R) \\
MustContain(I) \wedge IngGeneralizes(I, C) \wedge & \\
RecipeClasses(R, C) &\implies RecommendTarget(R) \\
MustNotContain(I) \wedge IngInRecipe(R, I) &\implies \neg RecommendTarget(R) \\
MustNotContain(I) \wedge IngGeneralizes(I, C) \wedge & \\
RecipeClasses(R, C) &\implies \neg RecommendTarget(R)
\end{aligned}$$

The first two rules let us consider a recipe based on the ingredient classes that it's composed of. The next set of rules let us use that information. We want to make recipes that contain ingredients in a user's query more likely. Furthermore, we also want to establish that any recipes that have ingredients that generalize to the same classes as an ingredient a user wants are more likely. The inverse goes for ingredients that a user does not want.

For adapting a retrieved case, we need to be a little creative. Because PSLs require all atoms implied by their predicates, we can't simply derive a probability for any two potential ingredients to swap. Considering only the 155 ingredients used in the CCC cocktail case library, we would need to infer probabilities on 155! predicates.

To get around this bottleneck, we consider two rules of thumb. It's best to perform the bare minimum number of swaps to satisfy a query and we only need to either swap in or out ingredients that are part of the user's query

Therefore, we inspect a user's query and generate the atoms in the Swap/2 predicate for each query. Each atom in Swap/2 contains a ingredient that the user has specified they wish to have or not have in the resultant recipe. We perform swaps with the following rules:

$$\begin{aligned}
MustContain(I1) \wedge RecommendTarget(R) \wedge IngGeneralizes(I1, C) \wedge & \\
IngGeneralizes(I2, C) \wedge IngInRecipe(R, I2) &\implies Swap(I1, I2) \\
MustNotContain(I1) \wedge RecommendTarget(R) \wedge IngGeneralizes(I1, C) \wedge & \\
IngGeneralizes(I2, C) \wedge IngInRecipe(R, I2) &\implies Swap(I2, I1)
\end{aligned}$$

Swaps are always read the same way, the first atom in the predicate is replacing the second. Due to the fact that the Swap/2 predicate is built on the fly, we don't need to specify a hard constraints that an element must be present. Probabilities are only inferred on the atoms present in the Swap/2 data files, and all of those atoms are related to a user's query. We then build the answer XML file based on the RecommendTarget/1 and Swap/2 predicates, as shown in figure 4.2.

The recommendation rules give us a set of probabilities on the Recommend-Target/1 predicates, however, unless the user was very, very specific with a query, several atoms are equally likely to fit. We take the set of the highest probable

```
RECCOMMENDTARGET(Gin_fizz_easy) = 0.877
RECCOMMENDTARGET(Cocktail_paradise) = 0.750
RECCOMMENDTARGET(The_kiss_of_Smurfette) = 0.994
RECCOMMENDTARGET(Cocktail_de_fruits) = 0.474
RECCOMMENDTARGET(Eggnog) = 0.000

SWAP(Apple_juice,Sour_cream) = 0.000
SWAP(Apple_juice,Pear_liqueur) = 0.750
SWAP(Apple_juice,Beer) = 0.891
SWAP(Apple_juice,Orange_syrup) = 0.750
SWAP(Apple_juice,Pineapple) = 0.000
```

Fig. 3. Some example values of a query looking for a cocktail containing Apple juce

atoms and chose one between them. We then retrieve that case from the CCC case library and use it as part of the retrieve half.

The adaptation rules give us a set of probabilities on potential swaps. We scan the Swap/2 predicate for the highest probability swaps that involve both the user's query and the retrieved recipe. We adapt the recipe by keeping swap quantity units and amounts the same, but changing the resultant ingredients.

5 Conclusion

We hope that ImpliedChef shows how PSLs can be used for CBR tasks like retrieval and adaptation. However, there are many extensions possible while using PSL as a framework.

The overview of PSL (and the rules that ImpliedChef uses) have kept to logical, boolean operations. PSL also supports arithmetic operations, such as sum constraints (the values of atoms need to sum to a particular value). PSL even affords substituting an arbitrary number of atoms as part of a logical rule with sum-augmented predicates, which work like placeholders for the sum of an arbitrary number of atoms. Select-statements restrict which atoms can be swapped in for sum-augmentation, so one can imagine encoding the amount of each ingredient in a recipe as a percentage, then using sum-augmentation and selection to be sensitive to ingredient amounts when adapting recipes.

In addition, PSL supports the use of arbitrary functions as part of implication rules, as long as those functions can take in string arguments and return a real value from $[0, 1]$.

$$Name(P1, N1) \wedge Name(P2, N2) \wedge Similar(N1, N2) \implies Same(P1, P2)$$

The above rule, for example, relates two people atoms (P) by their names (N). Similar/2 is a functional predicate, an external function that takes in two names and returns how similar they are from [0, 1]. This sort of technique allows for the unification of many ways to measure similarity, from WordNet comparisons to similarity metrics built from Long-Short Term Memory networks. More importantly, though, using several different rules with a variety of recommendation heuristics, we can tune the weights on the rules to fit a variety of user preferences.

In the current iteration of ImpliedChef, all of its atoms and rules come from the wikiTaaaable ontology. Other food ontologies exist that also have entities with labelings, such as the Foodon ontology [6]. Converting the relevant parts of other ontologies and integrating them into ImpliedChef is currently ongoing work.

Reflecting back on the opening problems, we can see that ImpliedChef shows how PSLs provide approaches for solving them. Recipe data and ontology information can be converted into PSL rules and predicates, and PSLs can be used for retrieval and adaptation CBR tasks. Furthermore, PSLs are able to reason over ingredient amounts, able to combine conflicting heuristic scores and able to pull in reasoning from multiple ontologies. They seem to be a powerful, general framework for tackling the semantically rich space of recipe generation.

References

1. Ahuja, J., Montville, J., Omolewa-Tomobi, G., Heendeniya, K., Martin, C., Steinfeldt, L., Anand, J., Adler, M., LaComb, R., Moshfegh, A.: Usda food and nutrient database for dietary studies, 5.0. US Department of Agriculture, Agricultural Research Service, Food Surveys Research Group (2012)
2. Bach, S.H., Broecheler, M., Huang, B., Getoor, L.: Hinge-loss markov random fields and probabilistic soft logic arXiv:1505.04406 [cs.LG] (2015)
3. Badra, F., Cojan, J., Cordier, A., Lieber, J., Meilender, T., Mille, A., Molli, P., Nauer, E., Napoli, A., Skaf-Molli, H., et al.: Knowledge acquisition and discovery for the textual case-based cooking system wikitaaable. In: 8th International Conference on Case-Based Reasoning-ICCBR 2009, Workshop Proceedings. pp. 249–258 (2009)
4. Delgrande, J.P.: An approach to default reasoning based on a first-order conditional logic: revised report. Artificial intelligence 36(1), 63–90 (1988)
5. Foulds, J., Kumar, S., Getoor, L.: Latent topic networks: A versatile probabilistic programming framework for topic models. In: International Conference on Machine Learning. pp. 777–786 (2015)
6. Griffiths, E., Brinkman, F., Dooley, D., Hsiao, W., Buttigieg, P., Hoehndorf, R.: Foodon: A global farm-to-fork food ontology
7. Hammond, K.J.: Chef: A model of case-based planning. In: AAAI. pp. 267–271 (1986)
8. Kouki, P., Fakhraei, S., Foulds, J., Eirinaki, M., Getoor, L.: Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems. In: Proceedings of the 9th ACM Conference on Recommender Systems. pp. 99–106. ACM (2015)
9. Morris, R.G., Burton, S.H., Bodily, P.M., Ventura, D.: Soup over bean of pure joy: Culinary ruminations of an artificial chef. In: Proceedings of the 3rd International Conference on Computational Creativity. pp. 119–125 (2012)
10. Renner, B., Sproesser, G., Strohbach, S., Schupp, H.T.: Why we eat what we eat. the eating motivation survey (tems). Appetite 59(1), 117–128 (2012)
11. Steptoe, A., Pollard, T.M., Wardle, J.: Development of a measure of the motives underlying the selection of food: the food choice questionnaire. Appetite 25(3), 267–284 (1995)

IntelliMeal - Enhancing Creativity by Reusing Domain Knowledge in the Adaptation Process

Kari Skjold, Marthe Øynes, Kerstin Bach, and Agnar Aamodt

Department of Computer Science
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
http://www.idi.ntnu.no

Abstract. This paper presents IntelliMeal, a case-based reasoning (CBR) system for recommending recipes. The main focus of the system is customizing recipes to a given user query re-using the domain knowledge of a CBR system within adaptation rules. In this work we implement a CBR system that works with a limited case base (21 recipes) and increase the amount of recipe recommendations by adapting these recipes using addition, creation, substitution, suitability and title name customization rules.

Keywords: Case-Based Reasoning, Rule Engine, Computational Creativity, Adaptation, Recipe Recommendation

1 Introduction

This paper presents IntelliMeal[1], a case-based recipe recommendation system addressing the open challenge of the 2017 Computer Cooking Contest (CCC). Since its initialization in 2008, the competition has been running almost every year with minor adjustments. Several research groups have contributed to the CCC over the years using information retrieval, information extraction and semantic technologies along with Case-Based Reasoning when developing recipe recommendation systems. The researchers have contributed with various approaches to the task. Four of the more influential systems are Taaable [2,4,6], CookingCAKE [5,9,10], JaDaCook [7,3], and CookIIS [8,11]. The Taaable researchers built their system around a collaborative, semantic Wiki, which also serves as the main knowledge base. The CookIIS researchers focused on the pre-processing of data to make the substitution of ingredients fluent and more realistic. CookingCAKE targets the preparation instructions by implementing cooking workflows, and lastly, the JaDaWeb researchers focused on the implementation of natural language understanding.

IntelliMeal is a knowledge engineering heavy system utilizing Case-Based Reasoning (CBR)[1]. The system aims to customize recipes for a given query consisting of desired and undesired ingredients. The main focus of the system

[1] www.intellimeal.no

is enhancing creativity in the adaptation process. The underlying goal of this paper is to investigate whether it is possible to build a CBR system that adapts recipes in a way that they satisfy a user's desires and expectations.

The case base consists of twenty-one sandwich recipe cases, hierarchical taxonomies and a set of adaptation rules. The taxonomies are separated into nine attributes or *ingredient categories*. Each taxonomy defines similarities between a restricted set of ingredients that belong to the given ingredient category. The taxonomies are extended versions of the taxonomies used in CookIIS [8,11], an earlier participant of the CCC. The attributes/ingredient categories are also used to construct each case and the ingredients included in the recipe.

The paper is structured as follows: Section 2 explains an adapted version of the CBR cycle [1], section 3 presents the evaluation of the systems in terms of similarity computation after adaptation and user evaluations of adapted recipes. In the final section we discuss our results and summarize our work.

2 Methodology and implementation

IntelliMeal implements all steps of the CBR cycle, but it includes a second retrieval phase after an ephemeral case base including temporarily adapted cases as been created. This allows IntelliMeal to assess the similarity of cases that have been modified based on desired and undesired ingredients specified in the query.

Figure 1 shows an overview of our implemented version of the cycle.

Step 1 The problem presented is the user query, which consists of desired and undesired ingredients. As figure 1 illustrates, the query is split in two: One undesired query containing the undesired ingredients and one desired query containing the desired ingredient.

Step 2 Our modified version of the CBR cycle involves *two* retrieval steps. The first process is the case base retrieval. It involves retrieving the cases from the case base with the highest similarity score to the user query. Hence, cases with the best *starting point* to end up in successful recipe recommendations. As the retrieval method employed in IntelliMeal is an important feature of the system, it is explained more detailed in section 2.1. The retrieved cases are further copied. The retrieved original cases are kept for later use while considering the copied versions in the reuse step.

Step 3 The reuse step is the most comprehensive step of the cycle. It is also the main focus of IntelliMeal. Hence, it is explained more detailed in section 2.3. The goal is to customize cases (i.e. recipes) so that they better fit the user query. However, with restrictions to avoid distasteful recipe results. As figure 1 illustrates, domain knowledge, rules, and the queries are used in the adaptation process. The result from the reuse step is adapted versions of the cases in the case base, further referred to as adapted instances.

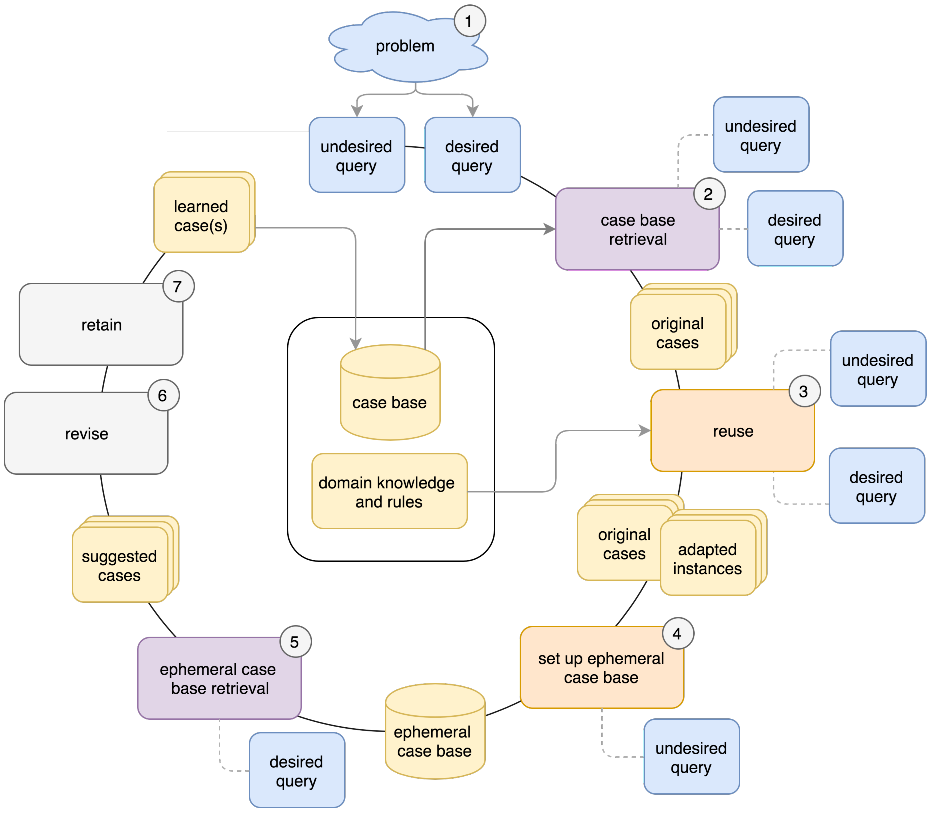


Fig. 1. Modified version of the CBR cycle

Step 4 The strategies used to set up an ephemeral case base mainly concerns using the undesired query to discard cases that are not satisfactorily. As figure 1 illustrates, both original cases from the case base as well as the adapted instances resulting from the reuse step takes part in the setup. The result is an ephemeral case base containing a selection of both types.

Step 5 The ephemeral case base retrieval involves comparing the cases in the ephemeral case base to the desired query. The result is a mixture of original cases and adapted instances, together with their resulting similarity score.

Step 6 The revision step involves getting feedback on recommendations. More specifically, the user gets the opportunity to confirm that adapted recipe recommendations are tasty.

Step 7 Only when an adapted recipe is confirmed tasty, the recipe instance will be added as a case to the case base together with the original cases. This refers to the retain step of the cycle.

2.1 Retrieval

Our retrieval method compares one query instance to all cases in a case base. The method considers one case at a time. It iterates through every attribute in the query and ignores attributes that are undefined. The rest of the attributes are considered valid and takes part in the similarity calculation.

For each valid attribute, the similarity between the query attribute and the corresponding case attribute is calculated by using the taxonomies. Then, they are weighted with a configured attribute weight. When calculating the total similarity between the query and a case, the attribute similarities are summed up and divided by the number of valid attributes. The retrieval method returns all cases ordered by their retrieved similarity score.

2.2 Rule engine

A rule engine was created for this specific system as a set of adaptation rules. There are two rule formats: *Simple Rule* and *Substitution Rule.*

If one rule were to specifically target only one ingredient, many rules would have to be written. Therefore, the rule engine is also able to consider all children of the ingredient. The children are fetched from the taxonomies. This enables one single rule to apply to hundreds of ingredients. For example, consider a rule saying any type of *meat* can substitute for any type of *fish.* These ingredients have 191 and 74 ingredients, respectively. Hence, this one rule will form 14 134 various combinations.

However, the functionality to ignore the children of a specific ingredients was also implemented. This can be used by writing a * after the ingredient. With this, a rule containing *meat** and *fish** would only form one combination.

Rule *requirements* refer to the ingredients in the recipe that have to be present for the rule to be valid. There can be zero or as many requirements as desired for a rule to fire. One requirement is satisfied if the recipe considered contains either the stated ingredient requirement or one of its children. A rule is valid when all requirements are satisfied.

Simple rules are the most basic rule type used in IntelliMeal. Based on given set of conditions (requirements req) an action is taken. For this rule type, all requirements have to be satisfied, before the rule can be applied.

$$req_1, ..., req_n \rightarrow ingredient \tag{1}$$

This rule type is used to create *addition rules*, *deletion rules*, *suitable rules* and *title rules.*

Substitution rules are more complex since they take both requirements and the ingredient to be substituted into account. Rules described by equation (2) takes the user requirement together with an ingredient to be substituted and the action describes which new ingredient can be included given a conditions, e.g. the existence or non-existence of other ingredients.

$$req_1, ..., req_n + ingredient_{old} \rightarrow ingredient_{new} + condition \tag{2}$$

An example for this rule is if the recipe contains tuna, any type of undesired supplements can be substituted out in favor for mayonnaise.

Furthermore, equation (3) shows a bidirectional rule that allows forward and backward substitutions:

$$req_1, ..., req_n + ingredient_{old} \longleftrightarrow ingredient_{new} \tag{3}$$

2.3 Reuse

In general, the reuse step involves three steps illustrated in figure 2: 1) Adaptation with the undesired query. 2) adaptation with the desired query and 3) suitable adaptation. The overall goal with the two first adaptation processes is to customize the retrieved recipes to better fit the user query. Suitable adaptation considers the modifications applied in the two first processes, examining whether more substitutions are necessary with the goal of making the recipe as a whole successful.

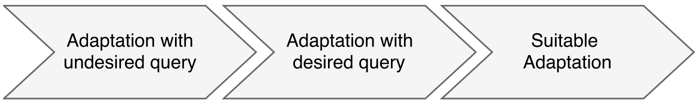


Fig. 2. Reuse process

Adaptation with the undesired query involves getting rid of undesired ingredients, while the goal with adaptation with the desired query is to add desired ingredients to recipes. These two adaptation processes are very similar. The difference is their opposite goals. Both processes loops through undesired/desired ingredients, respectively. For each ingredient, three substitution methods are considered: 1) Simple deletion/addition, respectively, 2) substitutions, and 3) similarity substitutions.

Deletion and addition rules are written in the *simple rule* format. When given requirements are satisfied, an ingredient may be deleted from/added to the recipe. Adaptation with the undesired query is carried out by deletion rules, while the adaptation with the desired query uses the addition rules.

Substitution rules are written in the *substitution rule* format. When some requirements are satisfied, a modification may be done to a recipe ingredient. That means, one ingredient is removed from the recipe and one ingredient is added. For adaptation with the undesired query, an undesired ingredient is removed and a *substitution alternative* is found. For adaptation with the desired query, a desired ingredient is added and a *substitution offer* within the recipe is found.

Similarity substitutions are the last type of substitutions considered. These substitutions are based on taxonomies. The system fetches ingredients that are similar to the ingredient that is considered. For adaptation with the desired query, the goal is to find an ingredient that is similar within the recipe, then remove this and add the desired ingredient. For adaptation with the undesired query, a similar alternative is added, and the undesired ingredient is removed. However, the substitution only goes through if an *adaptation threshold* is satisfied. A threshold is set for each ingredient category. It defines how similar the substitution alternative and the considered ingredient needs to be for the substitution to go through.

After the undesired and desired adaptation process, the style of the recipe may have changed. The idea with suitable adaptation is to make the new ingredients fit the recipe better. The process starts by iterating all modifications of the recipe instance. For every new ingredient, the system checks whether a so-called *suitable rule* applies. Suitable rules are of the type *Simple Rule*, which means that there may be requirements for the rule to be valid. Both addition rules, substitution rules, and similarity substitutions are considered in the process. If several suitable ingredients are found, a random one among them is chosen.

Suitable adaptation also involves modifying titles. When ingredients are replaced by new ingredients in a recipe, the recipe title may be out of context. Two types of adaptation methods were implemented to rename the title to better fit the new recipe. First, the system considers pre-defined rules. For the record, the rules are called *title rules* and are of the type *Simple Rule*. The rule may have requirements to be valid. If no specific rules on ingredients apply, a different approach is considered. The method focuses on the previous title, and apply to titles containing ingredients. The system aims to match the content of the title with the substitutions that are carried out. If substituted ingredients are found in the title, the same substitutions are carried out in the title.

3 Evaluation

Several evaluation methods were assessed to conduct the system's evaluation and three evaluation goals were set: 1) evaluate the calculated similarity score, and hence, the order of the suggested recipes, 2) show that adapted recipes better match the user query, and 3) evaluate whether the user can distinguish the computer created recipes from the human created recipes.

For the first goal, we compared the system's ranking of recipes with ten test subjects ranking of recipes, for three given queries. The difference was measured by calculating the difference between each recipes ranking by the system and its ranking by the test subject. Per query, this resulted in a score from 0 to 25, where 0 represents equal ranking, and 25 represents the opposite order between the system and the test subject's ranking. On average, the result of this process showed that the discrepancy in ranking between the test subject's and the system was 5.52 out of 25. Also, the test subject's and the system shared 3.57 out of 5 recipes as their top 5.

Fig. 3. Adaptation (green) changing the original similarity score (yellow)

		Predicted		
	n = 3414	FALSE	TRUE	
Actual	FALSE	TN 856 (50.09%)	FP 853 (49.91%)	1709
	TRUE	FN 911 (53.43%)	TP 794 (46.57%)	1705
		1767 (51.76%)	1647 (48.24%)	

Fig. 4. *Bot or not* user predictions vs. actual recipe creations

For the second goal, a three step process were conducted: 1) doing a set of queries with the adaptation process turned off and note the similarity scores achieved for the top five results, 2) doing the same set of queries with the adaptation process turned on, and 3) compare the similarity scores. The evaluation results showed that the average similarity score increases for all the test queries. Figure 3 illustrates the evaluation results. In the figure, the yellow bars show the average similarity score for the top five results with no adaptation, while the green bars show the average increment for the top five results with adaptation on. On average, the mean similarity score for the top five recipes suggested per query increased with 0.32.

For the third goal, an online quiz was implemented[2]. The quiz displayed one recipe at a time for the user, and the user was to guess whether the recipe was human or computer created. The quiz ran for seven days (168 hours) and gathered in total 3414 responses distributed over 42 recipes. Figure 4 shows a confusion matrix of the quiz feedback. To clarify, *true* refers to an adapted recipe. Results showed that people guessed that a human had created the computer adapted recipes in 53.43% of the cases. Also, people recognized the original cases as created by a computer in 50.09% of the cases. This result reveals that most users are not able to distinguish the recipes from one another.

4 Discussion and Conclusion

Earlier participants in the CCC have chosen various approaches for their systems. All four systems presented involve a hierarchical taxonomy. Some systems generate substitution rules from their taxonomy or cooking communities, while JaDaWeb has a table of ingredients that can substitute each other across categories. However, none explicitly define removal, addition or suitable rules like

[2] `www.intellimeal.no/botornot`

implemented in IntelliMeal. The rules and the similarity substitutions complement each other's weaknesses. Together, the components result in comprehensive adaptation of recipes. The rules employed in IntelliMeal have the ability to serve both specific and general purposes. Incorporating taxonomies into the rule engine has enabled this. Also, rules may be specified for ingredients across attributes. The result is more radical modifications of recipes, where the general style of a recipe may change completely.

In conclusion, IntelliMeal, extends the traditional CBR cycle by adding a second retrieval from an ephemeral case base populated with cases from a multi-layer rule engine. The adaptation mechanism gets creative in the way that it exploits domain knowledge defined in taxonomies to adjust similarity and to expand recipe case base. Therewith, we were able to achieve overall satisfying results with a limited case base. The measurable outcome of this project presented in section 3 is exceedingly satisfying. Adaptation of cases increased similarity scores for a given user query in all test cases, and humans had difficulties distinguish computer and human created recipes from one another.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. Artificial Intelligence Communications 7(1), 39–59 (1994)
2. Badra, F., Cojan, J., Cordier, A., Lieber, J., Meilender, T., Mille, A., Molli, P., Nauer, E., Napoli, A., Skaf-Molli, H., Toussaint, Y.: Knowledge acquisition and discovery for the textual case-based cooking system wikitaaable. Computer Cooking Contest Workshop Proceedings (2009)
3. Ballesteros, M., Martín, R., Díaz-Agudo, B.: Jadaweb: A cbr system for cooking recipes. Computer Cooking Contest Workshop Proceedings (2010)
4. Blansché, A., Cojan, J., Dufour-Lussier, V., Lieber, J., Molli, P., Nauer, E., Skaf-Molli, H., Toussaint, Y.: Taaable 3: Adaptation of ingredient quantities and of textual preparations. Computer Cooking Contest Workshop Proceedings (2010)
5. Fuchs, C., Gimmler, C., Günther, S., Holthof, L., Bergmann, R.: Cooking cake. Computer Cooking Contest Workshop Proceedings (2009)
6. Gaillard, E., Lieber, J., Nauer, E.: Improving ingredient substitution using formal concept analysis and adaptation of ingredient quantities with mixed linear optimization. Computer Cooking Contest Workshop Proceedings (2015)
7. Herrera, J., Iglesias, P., Sánchez, A., Díaz-Agudo, B.: Jadacook 2: Cooking over ontological knowledge. Computer Cooking Contest Workshop Proceedings (2009)
8. Ihle, N., Hanft, A., Althoff, K.D.: Extraction of adaptation knowledge from internet communities. Computer Cooking Contest Workshop Proceedings (2009)
9. Minor, M., Bergmann, R., Görg, S., Walter, K.: Adaptation of cooking instructions following the workflow paradigm. Computer Cooking Contest Workshop Proceedings (2010)
10. Müller, G., Bergmann, R.: Cookingcake: A framework for the adaptation of cooking recipes represented as workflows. Computer Cooking Contest Workshop Proceedings (2015)
11. Newo, R., Bach, K., Hanft, A., Althoff, K.D.: On-demand recipe processing based on cbr. Computer Cooking Contest Workshop Proceedings (2010)