Ontology Learning for Search Applications

Jon Atle Gulla, Hans Olaf Borch, and Jon Espen Ingvaldsen

Department of Computer and Information Science
Norwegian University of Science and Technology, Trondheim
jag@idi.ntnu.no

Abstract. Ontology learning tools help us build ontologies cheaper by applying sophisticated linguistic and statistical techniques on domain text. For ontologies used in search applications class concepts and hierarchical relationships at the appropriate level of detail are vital to the quality of retrieval. In this paper, we discuss an unsupervised keyphrase extraction system for ontology learning and evaluate its resulting ontology as part of an ontology-driven search application. Our analysis shows that even though the ontology is slightly inferior to manually constructed ontologies, the quality of search is only marginally affected when using the learned ontology. Keyphrase extraction may not be sufficient for ontology learning in general, but is surprisingly effective for ontologies specifically designed for search.

1. Introduction

Traditional ontology engineering approaches are tedious and labor-intensive, as the successful construction of high-quality ontologies requires a wide range of skill sets as well as an ability to deal with very complex and formal representations. The ontologies are expensive to develop and maintain, and it is often hard to manage and coordinate the contributions from various types of domain experts and ontology modelers. The subsea petroleum ontology developed by the Integrated Information Platform project, for example, currently contains more than 55.000 classes, has been constructed on the basis of existing ISO standards over 3 years in a 3 million Euro project and is still not ready as a new ISO standard [10]. At the same time, the ontologies are vital in Semantic Web applications, as they provide the vocabulary for semantic annotation of data and help applications to interoperate and people to collaborate.

Most ontology engineering methods today are based on traditional modeling approaches and stress the systematic manual assessment of the domain and gradual elaboration of model descriptions (e.g. [4,5]).

Ontology learning is the process of automatically or semi-automatically constructing ontologies on the basis of textual domain descriptions. The assumption is that the domain texts reflect the terminology that should go into an ontology, and that appropriate linguistic and statistical methods should be able to extract the appropriate concept candidates and their relationships and properties from these texts. Numerous approaches to ontology learning have been proposed in recent years

[7,10,11,14,15,17], and they seem to allow ontologies to be generated faster and with less costs than manual modeling approaches.

Even though many of the approaches display impressive results, the complexities of ontologies are so fundamental that the generated candidate structures often just constitute a starting point for the manual modeling task. Advanced approaches with deep semantic analyses of text or whole batteries of statistical tests tend to yield better results, but are expensive to develop and may still not compete with traditional ontology modeling with regard to its abilities to represent deep domain properties. However, the real quality of ontologies depends on its use in applications, its *application value*, which necessitates a consideration of how the ontology and the ontology engineering method match the requirements of the application.

Ontology-driven search applications use ontological structures to interpret and reformulate user queries. Only parts of the full ontology is useful to these applications, and the behavior of both the users and the domain collection may affect the way the ontologies should be constructed.

In this paper we present an unsupervised keyphrase extraction system that has been used to speed up the construction of search ontologies. The extracted keyphrases serve as concept candidates in the ontology and can even give indications for how hierarchical relations should be defined. This is a lightweight ontology learning approach, though cheap and practical to use for domains that evolve and lack available domain experts.

The paper is structured as follows. Section 2 discusses the required qualities of ontologies for search. We then introduce the keyphrase extraction system in Section 3 and briefly explain how it compares to manual ontology building based on a real case in Section 4. Section 5 introduces an ontology-driven search engine that uses ontologies to expand user queries. The semi-automatically generated and manually modeled ontologies are both plugged into the search application and evaluated with respect to search relevance in Section 6. Section 7 is devoted to related work, and the conclusions are found in Section 8.

2. Ontological Quality

Ontology-driven information retrieval incorporates a range of features that are not commonly found in traditional search applications. Commercial vector space search engines have started to make use of shallow linguistic technologies, but they do not attempt to expose the underlying semantics of documents [8]. An ontology-driven search application may in principle operate at three different levels of ambition. At the lowest level, we use concept hierarchies in the ontology to retrieve and present ranked documents to the user. The ontology is used to reformulate the query in terms of semantic concepts or to construct semantic indices. Slightly more challenging on the ontology side is the browsing of knowledge in the domain. The idea here is to let users explore relationships and hierarchies in the ontology to help him get an overview of the domain and find related information in a more interactive search session. At the most ambitious level, reasoning is employed to provide answers that are composed of several documents or implied by rules and axioms in the ontology.

A formally defined ontology language like OWL and a complete ontology with constraints and axioms must then be available [1]. Figure 1 illustrates how ontological information may be used in search.

Function	Focus	Ontology specification needed
Retrieve a document	Concepts	Concepts, hierarchies
Browse knowledge	Ontological structures	+ Properties, relationships
Compose a reply	Reasoning	+ logic, constraints

Fig. 1. Three applications of ontological information in information retrieval

As our research on search applications is on pure document retrieval, we will in this paper concentrate on the search quality of ontological concepts and hierarchies. The *ontology value quadrant* in Figure 2 is used to evaluate an ontology's usefulness in a particular application. The ontology's ability to capture the content of the universe of discourse at the appropriate level of granularity and precision and offer the users understandable and correct concepts are important features that are addressed in many ontology/model quality frameworks (e.g. [7,11,15]). But the construction of the ontology also needs to take into account behavioral aspects of the domain as well as the users of the application. For search ontologies, this means that we need to consider the following issues about content and behavior:

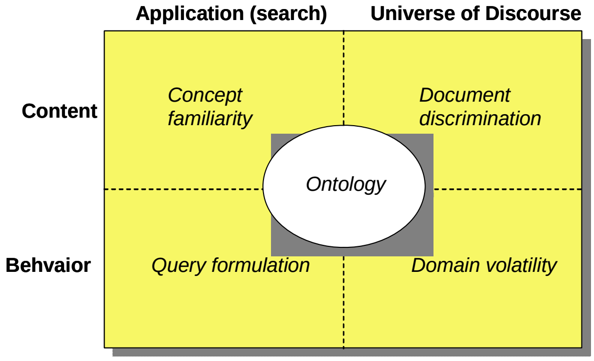


Fig. 2. Ontology value quadrant

- **Concept familiarity**. Terminologies are used to subcategorize phenomena and make semantic distinctions about reality. A high-quality ontology is made up of concepts that correspond to users' way of describing the same phenomena. Analyses of query logs reveal that users tend to use nominal phrases. Whereas we refer to user concepts not found in the ontology as *ignored concepts*, ontology concepts not appealing to users are called *superfluous concepts*.
- **Document discrimination**. The structure of concepts in the ontology decides which groups of documents can theoretically be singled out and returned as result sets. Similarly, the concepts implied in user queries indicate which groups of documents he might be interested in and which distinctions between

documents he considers irrelevant. If the granularity of the user's preferred concepts and the ontology concepts are compatible, combinations of these terms can single out the same result sets from the document collection. Result sets that can be implied by combinations of user-preferred concepts and not by combinations of ontology concepts are called *unfulfilled result sets*. Result sets that can be singled out by combinations of ontology concepts and not by combinations of user-preferred concepts are considered *superfluous result sets.*

- **Query formulation.** The user queries are usually very short, like 2-3 words, and hierarchical terms tend to be added to refine a query [8]. This economy of expression seems more important to users than being allowed to specify detailed and precise user needs, as very few use advanced features to detail their query. Hierarchical ontological structures corresponding to the users' query reformulation strategies are important.
- **Domain volatility**. Both the search domain itself and its documents may be constantly changing, and parts of the domain may be badly described in documents compared to others. The ontology needs to be constructed in such a way that regular and frequent updates are supported.

An ontology learning approach for search ontologies, thus, should be inexpensive and needs to generate familiar candidate concepts that enable the user economically to retrieve exactly those result sets that he might be interested in.

3. Ontology Learning with Unsupervised Keyphrase Extraction

Keyphrase extraction is the process of extracting an optimal set of keyphrases to describe a document. Whereas supervised keyphrase extraction employs a collection of documents with pre-assigned keyphrases to train the extraction algorithm, unsupervised extraction relies solely on a reference collection of plain unannotated textual data. Unsupervised keyphrase extraction has the advantage of being more widely applicable, since the method does not require any knowledge of the domain or consultation of domain experts. On the other hand, supervised keyphrase extraction normally produces more relevant keyphrases and can with repeated training improve the quality of its own keyphrases (see for example [18, 20, 22]).

A list of keyphrases gives a high-level summary of the document content. Such summaries can be used on search engine result pages, helping the user to decide which documents are relevant. It is also often used in document clustering or back-of-book index generation, though in this work we focus on ontology learning. Given a collection of documents describing a domain, the extracted keyphrases can be used to identify important concepts and provide a basis for constructing simple ontologies.

Figure 3 gives an illustration of the various steps in an unsupervised keyphrase extraction system for ontology learning. After cleaning and filtering the domain text, linguistic and statistical techniques are used to extract and rank candidate phrases from the domain. To avoid phrases that are common in all domains, we compare the candidates with a reference text and only include phrases that are characteristic to this particular domain. The final selection is done either by outputting a fixed number of keyphrases from each document in the collection, or selecting all keyphrases scoring higher than some threshold. The phrases may be single words, though usually the most interesting of them are longer noun phrases. After an appropriate set of candidate phrases has been identified, they are verified manually by domain experts and related to each other with various hierarchical and associative relationships. The multi-word keyphrases tend to give useful hints when constructing these hierarchies, but manual work is needed to complete the hierarchies and possibly add more abstract concepts that link everything together in complete ontologies.

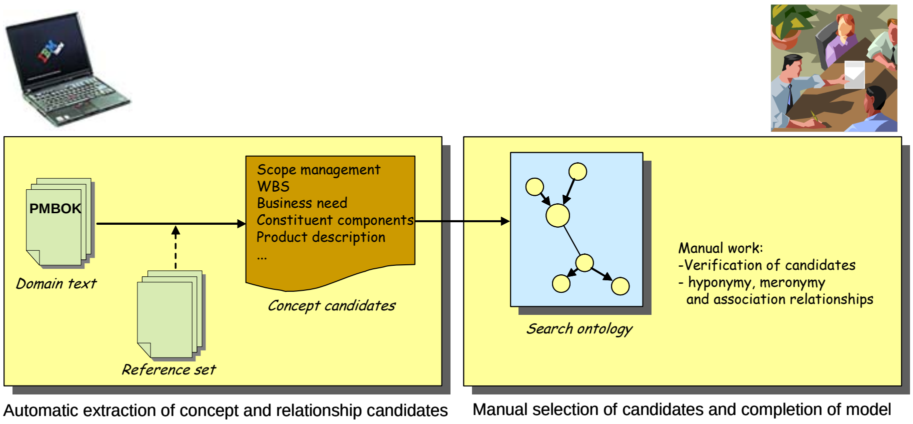


Fig. 3. Ontology learning with keyphrase extraction

4. Building Project Management Ontologies in STATOIL

STATOIL ASA is the leading petroleum company on the Norwegian Continental Shelf and has more than 25,000 employees in 31 countries. Most of their textual documents are structured in NOTES databases, but they are now in the process of implementing new applications and processes for information management. As part of this work, we are building ontologies that can enable ontology-driven search and more efficient application integration.

The domain chosen for the keyphrase extraction system was STATOIL's project management standard, PMI. This standard is enforced throughout STATOIL's organization and is well documented in books and reports. In particular, STATOIL is using a book called PMBOK[1] as a guide to people involved in projects. This book contains 12 chapters that define all the project terminology used in the management of STATOIL projects. We built two independent ontologies of the project management domain, one using keyphrase extraction and one with traditional modeling methods.

[1] Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK), 2000.

Semi-Automatic Ontology Learning

Our unsupervised keyphrase extraction system was first used to extract candidate concepts from PMBOK's 12 chapters. Each chapter in PMBOK was treated as a separate document, and all formatting and document structures were deleted. The resulting input to the extraction system was unannotated plain text, as shown by the PMBOK fragment below:

```
Scope planning is the process of progressively elaborating and
documenting the project work (project scope) that produces the
product of the project.
```

A Brill Part-Of-Speech tagger was then used to tag each word with its respective part of speech (POS):

```
Scope/NNP planning/NN is/VBZ the/DT process/NN of/IN progressively/RB
elaborating/VBG and/CC documenting/VBG the/DT project/NN work/NN (/(
project/NN scope/NN )/) that/WDT produces/VBZ the/DT product/NN of/IN
the/DT project/NN ./.
```

These POS tags come from the Penn Treebank tag set and allow us to filter out words that should not be considered potential keyphrases. Since our keyphrases should be composed of nouns, we concentrated on the words tagged with NN (singular or mass noun), NNP (singular proper noun) and NNS (plural noun). Stopwords were removed from the text, using a list of 571 words that are abundant in the English language and carry little or no discriminating meaning:

`Scope planning` **`is the`** `process` **`of`** `progressively elaborating` **`and`**
`documenting` **`the`** `project work (project scope)` **`that`** `produces` **`the`**
`product` **`of the`** `project.`

The words shown in bold were deleted from the text. To get rid of morpho-syntactic variation in the text, we used a lexicon to lemmatize the words. This means that the actual inflections are replaced by their corresponding base forms, giving us `plan` instead of the progressive `planning` and `produce` instead of the third person singular `produces`. If a word did not occur in the dictionary, Porter's stemming algorithm was applied to the word. This resulted in the following sequence of words (POS tags hidden):

```
Scope plan process progress elaborate document project work project
scope produce product project
```

Notice that the stemming of `progressively` to `progress` makes it appear like a noun, but we kept the tag RB to avoid that `progress` was analyzed as a noun later.

Different extraction systems tend to adopt different strategies for which structures should be considered potential keyphrases. In our system all consecutive nouns were selected as candidate phrases:

```
{scope planning, process, project work, project scope, product,
project}
```

The candidate phrases were weighted using the *tf.idf* measure used in information retrieval. We first calculated the term frequency (*tf*), which gave us an indication of how frequent this phrase was in this chapter compared to other phrases:

$$\mathrm{tf} = \frac{n_i}{\sum_k n_k}$$

where n_i is the number of occurrences of the considered phrase in the chapter, and the denominator is the number of occurrences of all terms (phrases) in the chapter. The total tf.idf score was calculated as shown below and takes into account the distribution of this phrase throughout the document collection:

$$\mathrm{tfidf} = \mathrm{tf} \cdot \log\left(\frac{|D|}{|(d_j \supset t_i)|}\right)$$

where $|D|$ is the total number of chapters in the collection and $|(d_j \supset t_i)|$ is the number of chapters (excluding the current chapter) where the term t_j appears (not equal to 0). The resulting list of weighted phrases were sorted and presented to the user:

```
{(scope planning, 0.0097), (project scope, 0.0047), (product,
0.0043), (project work, 0.0008), (project, 0.0001), (process,
0.0000)}
```

A total of 180 keyphrases, 15 for each chapter of PMBOK, were selected. These were simply the 15 top-ranked phrases of each chapter, based on the tf.idf score. A domain expert from STATOIL was then asked to mark out those keyphrases that would not be suitable as ontological concepts. With these phrases removed, we had 106 phrases left that were manually structured as an ontology. Synonyms were identified, and the appropriate hierarchical relations were added manually to form a full ontology.

The resulting ontology, which was represented in OWL, contained 3 hierarchical levels, 106 concepts (classes) and 6 synonyms.

Manual Ontology Construction

We also constructed a project management ontology manually. The modelers were familiar with the PMI standard in STATOIL, had access to PMBOK, and also had some experience in running small projects using similar methodologies.

The manual modeling process was substantially longer than the semi-automatic ontology learning process. It led to a larger ontology, with deeper structures, more concepts and more synonyms. The manually constructed ontology had 5 hierarchical levels, contained 142 concepts and 26 synonyms.

5. Ontology-Driven Search

For the evaluation of our ontologies' application value, we installed an ontology-driven search application that uses ontologies to interpret and expand user queries. The idea is to add weighted synonyms and semantic relations to retrieve relevant documents that do not necessarily contain the search terms per se. This is an approach that tends to increase recall rather than precision, which is often preferable for such a small domain with a limited number of indexed documents available.

Take for example the query *'human resource'*. The original query terms as well as their synonyms get weight 1.5. Expanding this query with synonyms and semantically related concepts with slightly lower weights, we get the following two reformulated queries for the two ontologies at hand:

Expanded query with manual ontology	**Expanded query with automatic ontology**
'human resource' (1.5), hr (1.5), 'organizational planning' (1.0), staff (1,0), 'staff acquisition' (1.0), 'team development' (1.0)	*'human resource' (1.5), 'human resource management' (1.0), 'organization chart' (1.0), role (1,0), chart (1.0), staff (1.0), 'staff assignment' (1.0), 'team competencies' (1.0), 'team development' (1.0)*

The term *hr* is given the same weight as *'human resource'*, as it is considered a synonym in this domain. All the other terms are related to 'human resource' through associations or abstractions and are given a weight of 1.0. The different expansions for the two cases reflect the differences of the two ontologies. As seen from this example, the semi-automatically generated ontology has found more semantic links between human resource and other concepts. On the other hand, only the manually built ontology includes the synonym *hr*.

The expanded weighted query is the system's interpretation of the user's real information needs. After mapping the query onto corresponding search terms, a standard vector model based search engine (Lucene) is used to retrieve and rank documents relevant to the new query.

6. Search Quality of Ontologies

The ontologies were first evaluated independently of their applications. Domain experts from STATOIL ranked the ontology concepts with respect to their suitability in a real full-fledged project management ontology. Since the manually constructed ontology was larger, it was not surprising that it also contained more relevant domain concepts. This ontology had 122 very good concepts against the other ontology's 73 very good concepts, which means that that manual process had managed to uncover 67 % more high-quality ontology concepts. If we take the total number of concepts into account, though, the difference of quality is not so great. Whereas in the manual process around 86 % of the concepts were considered to be of high quality, about 69 % of the semi-automatically generated concepts were of the same quality. For an equal number of concepts, thus, we may conclude that the manual process would give us slightly less than 25 % more high-quality concepts.

It could be tempting to improve the semi-automatic ontology by extracting more keyphrases than the 180 we extracted in this experiment. However, it turns out that the quality of keyphrase extraction is highly dependent on the size of documents available to the analysis. As shown in Figure 4, we need documents of at least 5-6,000 words to get an R-precision of more than 0.4 when the top 15 phrases are included, and it seems very difficult for this method to reach 0.6 for even very large documents. For chapters 1, 3, 4, 9 and 10 it would quality-wise have been better to extract fewer than 15 phrases. Chapter 2 does not follow the general trend of getting better phrases with longer documents, as it deals with the context of project management and the extracted terms were considered out of scope by the experts.

A practical evaluation of the ontology-driven search applications was then run on a document index from STATOIL that contained rather small project management and project-related documents. Two separate search applications were set up to work on the same index. Whereas one application used the manually constructed ontology to interpret and reformulate queries, the other one made use of the ontology constructed with the help of our keyphrase extraction system.

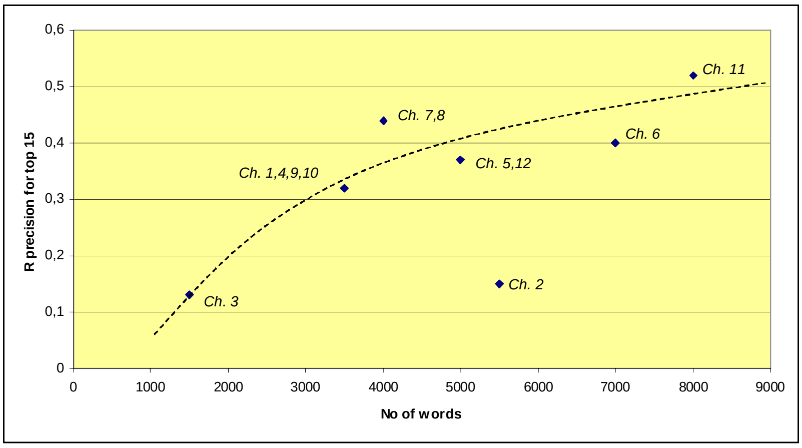


Fig. 4. R-precision and chapter size for extracted ontology concepts

As the intention was to evaluate and compare ontologies rather than to evaluate the search application itself, we defined a total of 16 queries that were all related to concepts in the two ontologies. The expansions of these queries would naturally differ, as the concepts were modeled differently and related to different concepts in the ontologies.

A group of six people were asked independently to run the queries on the two search applications and rate the top 5 documents for each query from 0 (not relevant) to 2 (highly relevant). The total score for one individual's evaluation of one query for one application was given as:

$$Q = 1/2 * \sum_{i=1}^{5} S_i * W_i$$

where S_i is the individual rate of document $D_{i,}$ and W_i is a weight that ranges from 10 for the top ranked document to 1 for the 5th ranked document. After combining the results from each of the six individuals and normalizing the average scores for all queries for the two search applications, we got the results shown in Figure 5(a).

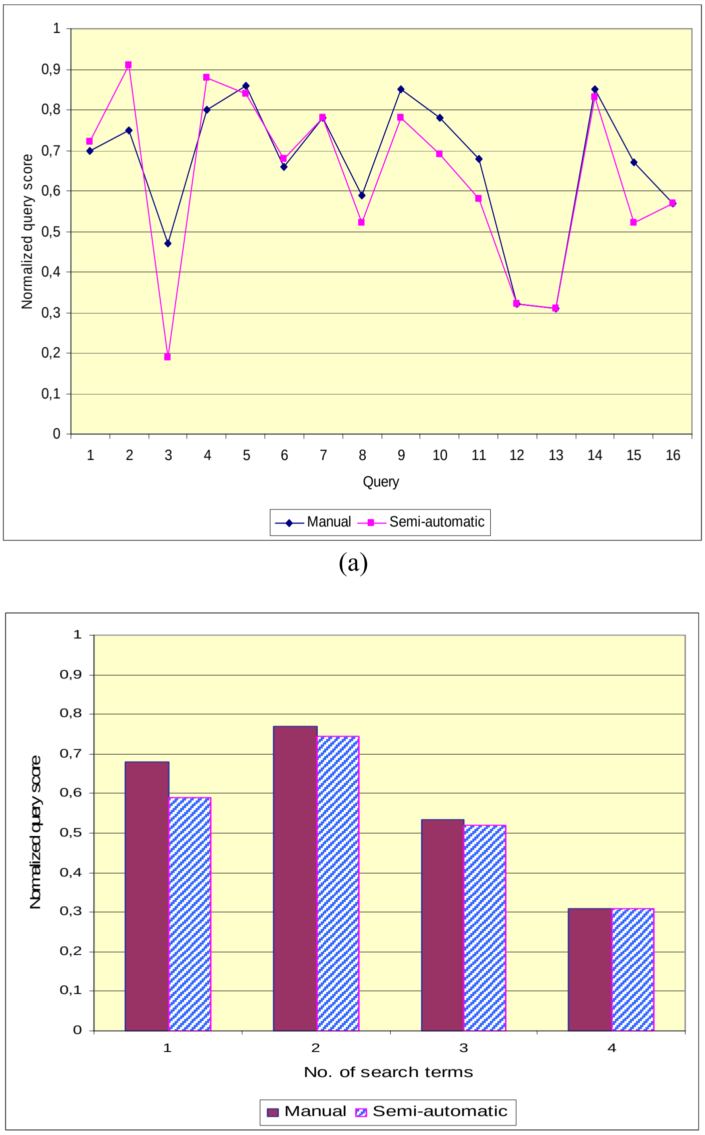
The search application performs slightly better with the manually constructed ontology. In 50% of the queries the manual ontology wins out, and only 25% are answered better with the generated ontology. However, the score differences are in most cases very small. On the average, the query scores for the manual ontology are only 5.1 % higher than for the generated ontology. Taken into account that the manual ontology had 67 % more high-quality concepts and a ratio of good to neutral ontology concepts almost 25 % higher than for the generated ontology, this difference is surprisingly small. It seems that the search quality of ontologies is not so dependent on an exact match between ontological concepts and domain experts' judgments, as long as they are reasonably well defined with respect to the documents available in the domain.

Another interesting observation is illustrated in Figure 5(b). If we group the results on the basis of number of query terms, we can easily see where the two search applications differ in quality. For queries that deal with one-term concepts, like *procurement* and *stakeholder*, the manual ontology performs substantially better than the semi-automatic one. For long detailed queries, like '*cost performance index*' and '*work breakdown structure*', there is practically no difference between the two ontologies.

This seems to support that the manual ontology contains better concepts and relationships between concepts. Presumably, the importance of good concepts and well-defined relationships are more important when the query is short and vague by itself. For longer queries, the user has already specified his information needs so accurately that the addition of related terms may not contribute much. This may also indicate that ontology-driven query expansion in general has only limited effect when queries are precise and unambiguous.

7. Related Work

In this paper we have investigated to what extent unsupervised keyphrase extraction may be useful in speeding up the construction of ontologies for search applications. Our idea of taking the intended use of the ontologies into account is not new. Thurmair claims that precision and recall are useless in keyphrase extraction, and the quality of extracted terms must be assessed on the basis of how people make use of the terms and how fast they can define their own term subsets [15]. Tomokiyo and Hurst propose an unsupervised extraction strategy based on n-grams, and they require that the users themselves characterize what constitutes proper phrases for their particular applications [17].



(a)

(b)

Fig. 5. (a) R-precision for queries. (b) R-precision as function of number of query terms

One of the most well-known workbenches for ontology learning is Text2Onto, which includes a whole battery of statistical and linguistic text mining components [3]. Text2Onto is meant to support a wide range of analyses and has a flexible and exapandable architecture. This modular approach to text mining is also adopted in other applications [7,10]. As opposed to these workbenches, our system is more lightweight and tailored to the restricted need in constructing and maintaining search ontologies.

OntoLT in Protégé includes traditional statistical methods for term extraction, though its main contribution lies in the use of shallow linguistics to extract structured information from individual sentences [2]. It uses a rule-based system for German and English sentence analysis, SCHUG, to propose properties and relationships based on the recognition of heads, modifiers and predicates in the sentences. A similar approach to linguistic sentence analysis is adopted by Sabou et al. to extract concepts and relationships between concepts in a web service context [14]. These methods are able also to suggest relationships between concepts, but it is an open question how this sentence by sentence approach will work for large text collections where individual sentences are statistically insignificant and aggregated data need to be used to produce representative results.

Our search application had a rather simplistic approach to query expansion. As noted by Voorhees, it is not obvious that adding semantically related terms will improve the quality of the search application [21]. However, experiments with domain-dependent vocabularies – instead of Voorhees' WordNet approach – does indicate that careful semantic refinement of queries may be useful [18]. Mitra et al. [13] is refining the query based on *blind feedback*, i.e. the system itself selects documents that are considered relevant to the original query and uses these documents to construct an expanded query without any human involvement. Similarly, detecting word relationships from result sets and using these to expand the original query with related terms has been tested successfully by for example Xu & Croft [23]. Interestingly, their text mining approach to query expansion has many similarities with our approach using automatically generated ontologies. We apply text mining to construct ontologies off-line, and these ontological structures are afterwards used to expand the queries. A fundamental difference is that our text analysis is done on the whole document collection, whereas their analysis only makes use of documents considered relevant to the unexpanded query.

8. Conclusions

Unsupervised keyphrase extraction is a flexible and inexpensive method for generating candidate concepts to search ontologies. They do not require any particular preparation or involvement of domain experts and are thus well suited to unstable domains like document collections. Using tf.idf to rank keyphrases, we also end up with phrases that are well suited to single out documents in the collection.

The quality of extracted keyphrases is not at the same level as for supervised extraction, though their quality increases with the size of the documents used in the process. It is clear that the keyphrase extraction-based ontology learning method will not produce as many high-quality domain concepts as the manual approach. However, when applied as a search ontology, the quality of the search application is not much affected if a generated ontology replaces a manually built one. For the application value of the search ontology, it seems equally important that the ontology is well adapted to the document collection as that the concepts perfectly model the domain itself. There is a trade-off between the costs of developing and maintaining high-quality ontologies and the benefits of using them in ontology-driven applications.

Unsupervised keyphrase extraction is a promising approach to search ontology engineering, though there are still many aspects of search ontologies that this approach as well as other approaches do not address properly. A good search ontology is specified at a level of granularity that corresponds to the needs expressed in user queries. It should contain concepts that are familiar to the users and allow him to express his information needs in an economic and efficient way. However, we cannot restrict the user to only use already defined concepts and we need a way to interpret user queries that involve non-concept terms that may or may not be related to ontological structures.

References

1. Antoniou, G.; Franconi, E.; van Harmelen, F. Introduction to Semantic Web Ontology Languages. In Eisinger, N. and Maluszynski, J. (Ed.), Reasoning Web, First International Summer School 2005, Chap. 1, Malta, July 2005. Springer.
2. Buitelaar, P., Olejnik, D., Sintek, M.: A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. In: Proceedings of the 1st European Semantic Web Symposium (ESWS), Heraklion, Greece, May 2004.
3. Cimiano, P., Völker, J.: Text2onto – A Framework for Ontology Learning and Data-Driven Change Discovery. In Proceedings of 10th International Conference on Applications of Natural Language to Information System (NLDB 2005), Alicante, June 15-17, 2005, pp. 227-238.
4. Cristiani, M., Cuel, R.: A Survey on Ontology Creation Methodologies. Idea Group Publishing. 2005.
5. Fernandez, M., Goméz-Peréz, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. In Proceedings of the AAAI'97 Spring Symposium Series on Ontological Engineering, pp 33-40, Stanford, 1997.
6. Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. Nature Genet. Vol. 25, pp 25-29. 2000.
7. Goméz-Peréz, A.: Evaluation of ontologies. International Journal of Intelligent Systems, Vol. 16, No. 3, pp 391-409, 2001.
8. Gulla, J. A., Auran, P. G., Risvik, K. M.: Linguistics in Large-Scale Web Search. In Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems (NLDB 2002), Stockholm, June 2002, pp. 218-222.
9. Gulla, J. A., Brasethvik, T., Kaada, H. A Flexible Workbench for Document Analysis and Text Mining. In Proceedings of the 9th International Conference on Applications of Natural Language to Information Systems (NLDB'04), pp. 336-347, Manchester, June 2004. Springer.
10. Gulla, J. A., Tomassen, S. L., Strasunskas, D.: Semantic Interoperability in the Norwegian Petroleum Industry. Submitted to the International Conference on Information Systems and Its Applications (ISTA'06), 2006.
11. Lindland, O. I., Sindre, G., Sølvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Software, Vol. 11, No. 2, pp 42-49, March 1994.
12. Maedche, A. Ontology Learning for the Semantic Web. Kluwer Academic Publ. 2002.
13. Mitra, M., Singhal, A., and Buckley, C.: Improving Automatic Query Expansion. In Proceedings of 21st annual international ACM SIGIR conference on Research and Development in Information Retrieval, 1998, pp. 206-214. ACM Press.

14. Navigli, R., Velardi, P.: Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. Computational Linguistics, Vol. 30, No. 2, pp. 151-179. June 2004.
15. Pinto, H. S., Martins, J. P.: Ontologies: How can They be Built? Knowledge and Information Systems, Vol. 6, No. 4, pp 441-464, July 2004.
16. Sabou, M., Wroe, C., Goble, C., Stuckenschmidt, H.: Learning Domain Ontologies for Semantic Web Service Descriptions. Accepted for publication in Journal of Web Semantics.
17. Thurmair, G.: Making Term Extraction Tools Usable. The Joint Conference of the 8th International Workshop of the European Association for Machine Translation and the 4th Controlled Language Applications Workshop (EAMT/CLAW'03). Dublin 2003.
18. Tomassen, S. L., Gulla, J. A., Strasunskas, D.: Document Space Adapted Ontology: Application in Query Enrichment. In Proceedings of 11th International Conference on Applications of Natural Language to Information Systems (NLDB 2006), Klagenfurt, 2006, pp. 46-57. Springer Verlag.
19. Tomokiyo, T.; Hurst, M.: A language model approach to keyphrase extraction. In Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisitions and Treatment, 2003.
20. Turney, P. D.: Learning algorithms for keyphrase extraction. Information Retrieval, Vol. 2, No. 4, pp 303-336, 2000.
21. Voorhees, E. M.: Query expansion using lexical-semantic relations. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, 1994, pp. 61-69. Dublin. Springer-Verlag.
22. Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., Nevill-Manning, C. G.: KEA: Practical automatic keyphrase extraction. In ACM DL, pp 254-255, 1999.
23. Xu, J. and Croft, W. B.: Query Expansion Using Local and Global Document Analysis. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1996, pp. 4-11. Zurich. ACM Press.