A Scalable Framework for Data-Driven Ontology Evaluation

Daniel Knoell[1], Martin Atzmueller[2], Constantin Rieder[1], and Klaus Peter Scherer[1]

[1] Karlsruhe Institute of Technology
D-76344, Eggenstein-Leopoldshafen, Germany
firstname.lastname@kit.edu

[2] University of Kassel, Research Center for Information System Design
Wilhelmshöher Allee 73, 34121 Kassel, Germany
atzmueller@cs.uni-kassel.de

Abstract. Ontology evaluation is an important method supporting the lifecycle of ontology engineering. In particular, in the context of self-learning support systems that aim at supporting ontology learning the evaluation of the learned ontology is crucial. In order to provide widespread application, e.g., in industrial use cases and Big Data contexts, there are also challenges with respect to scalability that existing solutions do not explicitly address. Therefore, this paper presents a scalable data-driven framework for ontology evaluation, especially targeting Big Data scenarios and use cases. We provide a first instantiation of the framework, discuss first experiments and report on promising results.

1 Introduction

Ontology evaluation is an important step in the lifecycle of ontology engineering, specifically in development and maintenance. Ontology evaluation can be described as the "assessment of the quality and the adequacy of an ontology or parts of it regarding a specific aim, goal or context" [13]. Thus, for supporting the process of ontology engineering appropriate approaches and tools are necessary in order to support these assessment aspects for checking the quality of an ontology.

For supporting such processes, we aim at a data-driven approach, e. g., in order to answer such questions such as how well the ontology fits a given corpus, or which extensions are necessary. While there are existing approaches targeting similar issues, we specifically focus on a scalable approach in order to allow the application in Big Data contexts, i. e., for the scalable processing of large amounts of data such as efficient self-learning support systems. These systems guide the user, for example in the form of an expert system, through complicated problems. Self-learning procedures for the development and maintenance of a knowledge base, necessary for most of the support systems, are already in use [12].

Our contribution can thus be summarized as follows: We provide a scalable framework for data-driven ontology evaluation and present first instantiations of the framework in the context of large datasets. Furthermore, we discuss first experiments using these also focusing on the scalability of the proposed approach.

The rest of the paper is organized as follows: Section 2 discusses related work. After that, Section 3 describes the proposed scalable framework for data-driven ontology evaluation. Next, Section 4 discusses first results instantiating the presented framework. Finally, Section 5 concludes with a summary and interesting directions for future work.

2 Related Work

There are several approaches for data-driven ontology evaluation, e. g., [8]. Brewster [9], for example, describes two basic and a more sophisticated comparison method. The first basic method is an automated term extraction step where the number of overlapping terms between the ontology concept labels and the extracted terms are counted. The second method uses a vector space representation of the terms to get an overall measure of fit. The more sophisticated approach first identifies the keywords via automated term recognition. Then the query expansion adds two levels of hypernyms by using Wordnet or IR techniques. In the last step, the identified terms are mapped to the ontology. This approach of Brewster embracing comparisons with data about the domain to measure the coverage with the ontology outlines a simple option of instantiating our framework. Compared to that approach, however, our framework provides a much more scalable architecture and more sophisticated and flexible options for instantiation.

The correctness of ontologies are measured against a corpus of documents about the domain. These ontologies are based on domain knowledge which is dynamic and changes over different dimensions (e. g., temporal, categorical). Hlomani [14] explores how multiple dimensions affect the results of data-driven ontology evaluation presenting a theoretical framework and metrics that account for bias along the dimensions of domain knowledge. Our approach adopts parts of Hlomani's theoretical results. As demonstrated by our first experiments, we extend our architecture with interfaces and components in order to support the processes with Big Data technologies, in combination with adaptations and extensions of the methods of Brewster outlined above.

In [18], Maedche and Staab investigated ontologies as two-layered systems describing a methodological inventory to measure an overlapping and fitting of two ontologies at a lexical and a conceptual level. In addition to creating a methodical baseline, practical experiences were collected for its applicability. In contrast to this approach using comparisons to a gold-standard-ontology with the focus on tasks such as detecting and retrieving relevant ontologies to support ontology creation our investigation concentrates on data-driven ontology evaluation utilizing Big Data techniques.

A related corpus-based evaluation approach is proposed in [22]. The presented method might in principle be applied to a comparison of an ontology with either an expert-provided reference, or with a generated list of related terms. However, the proposed results are assigned to manual linguistic evaluation and the focus lies on measuring which terms fit best for an ontology at the time of its creation by using four defined measures. This approach could potentially be another instantiation of our framework.

To the best of the authors' knowledge, currently there is no comprehensive framework which aims for data-driven ontology evaluation in a Big Data context. Our proposed solution focuses on a Big Data approach to ontology analytics integrating a framework that enables to support ontology learning and evaluation at large-scale.

3 Description of the Framework

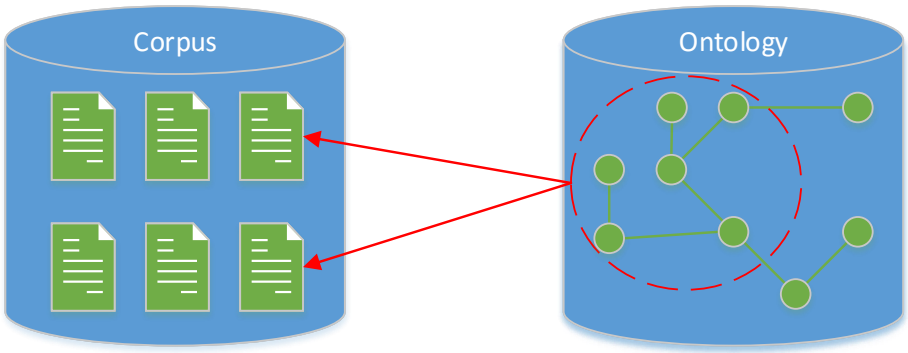

Fig. 1: Overview of a typical problem setting.

The starting point of the proposed framework is shown in Figure 1: A corpus with text documents and an ontology, or a list of words (in the following only ontology). The ontology should be used for the semantification of the corpus. This is necessary for applications like semantic search or semantic navigation. For the semantification of a corpus, it is then important to know that the corpus and the ontology fit together. In some cases we can leave out parts of the ontology, which are not important for our corpus. In other cases we need to add parts to the ontology so it can be used with this corpus. This is outlined in Figure 1, where only the circled part of the ontology fits to the corpus. With the framework we want to address such issues, i. e., in order to rate how good the ontology fits to the corpus and also to identify unnecessary and missing elements. We also work with huge corpora, so it is necessary to design the framework in a modular and scalable fashion. The modularity is important to be flexible with the setup. On the one hand, if we only have a small dataset and hardware with limited performance (like a standard laptop), we need the possibility to easily adapt the framework to these circumstances and use only a single computer with the local file system. On the other hand, if we work with Big Data and have the opportunity to use a computer cluster, we want to use the computing power of the whole cluster utilizing, for example, a Big Data storage mechanisms like the Hadoop Distributed File System (HDFS).

An overview of the framework is shown in Figure 2. It starts with the preprocessing where the data is converted in a suitable format and a process for getting the important terms out of the corpus (term extraction) is implemented, e. g., using rule-based methods [4]. Even in this first step a Big Data framework like Apache Spark or Hadoop can be applied. Then an attributed multiplex graph is generated, modeling and capturing relations between the nodes in potentially different layers, and modeling properties of those by labels assigned to nodes and/or edges. How this is done in detail depends on the methods used.

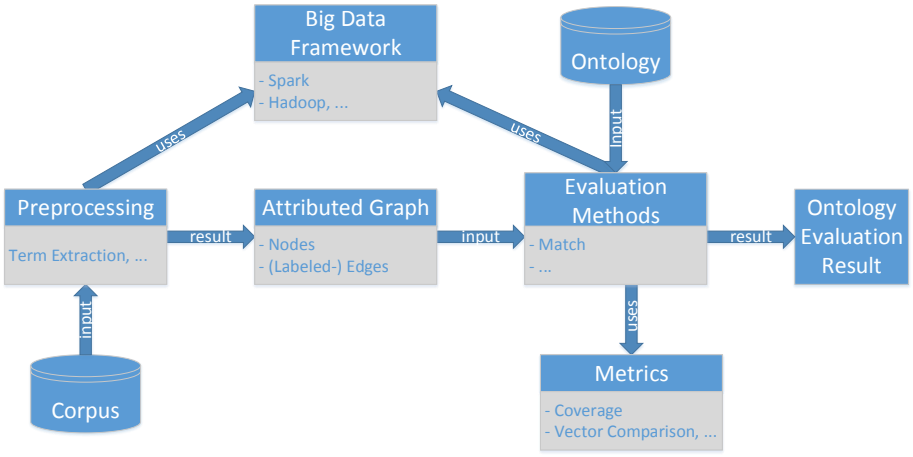

Fig. 2: Overview on the proposed framework for data-driven ontology evaluation.

One exemplary instantiation applies terms from the term extraction as nodes. The next step creates relations between the extracted terms by using the structure of the sentences. These relations can then form the edges for the graph. To get labels for the edges a semantic analysis of the sentences is needed. This analysis can be done, e. g., with a tool like JoBimText [7]. When the attributed graph is built, it can be compared to the ontology. Here, for example, quality measures and metrics, like coverage or vector comparison, can be applied, also including adapted techniques for assessing (semantic) similarity and grounding, e. g., [11, 17, 19]. The evaluation methods can then also be implemented in a Big Data environment like Spark or Hadoop in order to allow a scalable process for deriving the ontology evaluation result.

In the outlined application example, there are two possible results for this process, depending on the specific application. The first one shown in Figure 3 is a clustered quality assessment: The user can see which part of the corpus fits best to the ontology. The clusters of the corpus and a quality measurement are shown, so the user can see how good the cluster matches with the ontology. This result is interesting for users who wnat to extend an existing ontology, by adding concepts and relations using the corpus. With the help of this assessment, the user can decide if he wants to use the whole corpus for ontology learning or only specific parts, which fit the best to the initial ontology.

The second possible result, displayed in Figure 4, is the proposed ontology extension for corpus partition. This is important for users aiming at an ontology which to be fit to their (specific) corpus. If the user does not have an ontology for a specific corpus, but is able to identify an ontology for a related topic, then this ontology can be potentially adapted. For that, the framework needs to be appropriately instantiated: In this example it adds relations and concepts, which occur in the corpus, to the initial ontology (lower part of Figure 4) and deletes concepts and relations, which are not available in the corpus (upper right part of Figure 4). So at the end the user has a suitable descriptive ontology and can use it for further ontology learning together with the corpus.

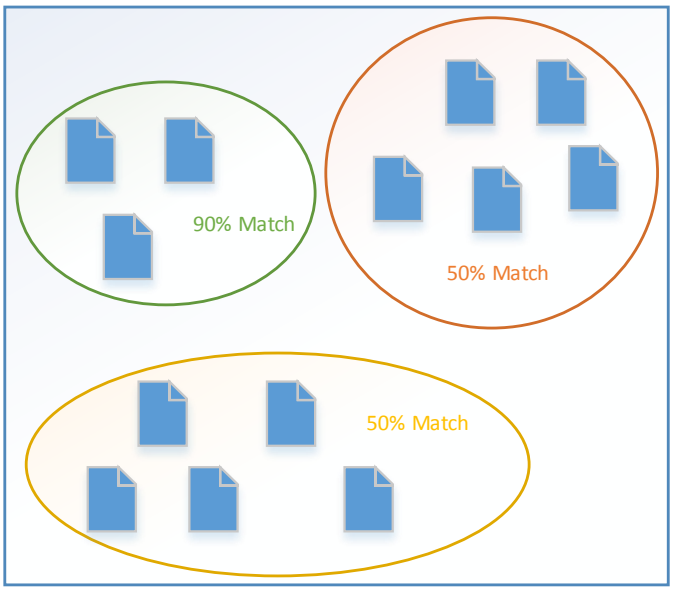

Fig. 3: Result: Quality assessment.

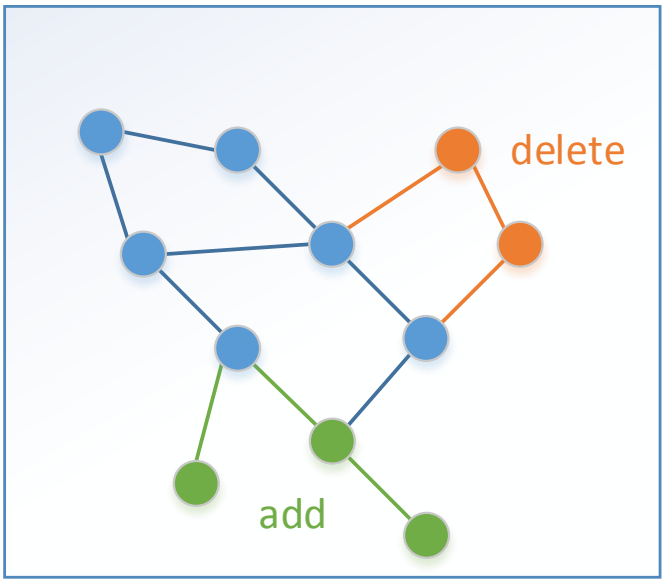

Fig. 4: Result: Ontology extension.

4 First Experiments

To instantiate the framework we first focus on a proof-of-concept approach, which is based on simple text similarity checks, also relating to the first step in our framework. It is an adaptation of the the basic approach from Brewster [9]. This step is mainly a proof of concept for the scalability and modularity of the concept. Therefore, we use an automated term extraction step to get the nodes of the attributed graph and calculate the overlap of the nodes with the concepts of the ontology. We use the coverage for estimating how good the ontology fits to the corpus. This approach will be tested not only on single computers but also on a computing cluster infrastructure. Furthermore, we will sketch the instantiation of the framework in a more sophisticated approach.

4.1 Proof-of-Concept Instantiation of the Framework

At the current state of the framework, the first approach was implemented and run on a single core computer. As metric we used the coverage of the extracted terms [15]. In contrast to Brewster [9] we can both handle simple keyword lists as well as (complex) ontologies as input. Furthermore, our framework is designed to enable large-scale data processing handling Big Data. This is achieved by using Apache Spark as engine, which also allows reading data from a HDFS.

In our case, the corpus is a sample of scientific papers from arxiv.org. They deal with topics like Astrophysics, Condensed Matter, General Relativity and Quantum Cosmology or High Energy Physics. The full-text access is provided via Amazon S3 [6] and available as pdf and source file access [10]. The source file are mostly TeX/LaTeX with figures. For the word-list, the Institute of Electrical and Electronics Engineers (IEEE) thesaurus is used. It is a controlled vocabulary of over 9,700 descriptive engineering, technical and scientific terms, as well as IEEE-specific society terms[15]. The choice for these two datasets was made because they both include scientific terms, but are not exactly from the same domain. That is why we hope to get some good suggestions for changing the ontology, like already described in Figure 4, in the future. If the corpus and the word-list would fit perfectly together, there could no suggestion for optimising the ontology be made. At this point the calculation of clusters and the ontology extension is not implemented yet. In this first proof-of-concept approach we particularly stress the scalability of our solution. The framework is designed to be modular, so every component can be exchanged; Apache Spark was used for large-scale data processing.

The input format of the corpus and the IEEE thesaurus is PDF. So in the first step the PDFs were converted to plain text, using the linux tool “pdftotext”, and unnecessary descriptions were deleted. Using the LaTeX source files of arXiv provided worse results than using the pdfs. After that the preprocessing of the corpus, depicted in Figure 5, starts. The documents were split up into lower-case words and stop words, special characters and small words with less than three characters were removed. This is performed in a stream on a Spark data structure. Then a map-reduce word count is performed and only words with a frequency higher than 1000 were used for the comparison with the word-list. For future work, this threshold could be made relative and independent of the size of the corpus.

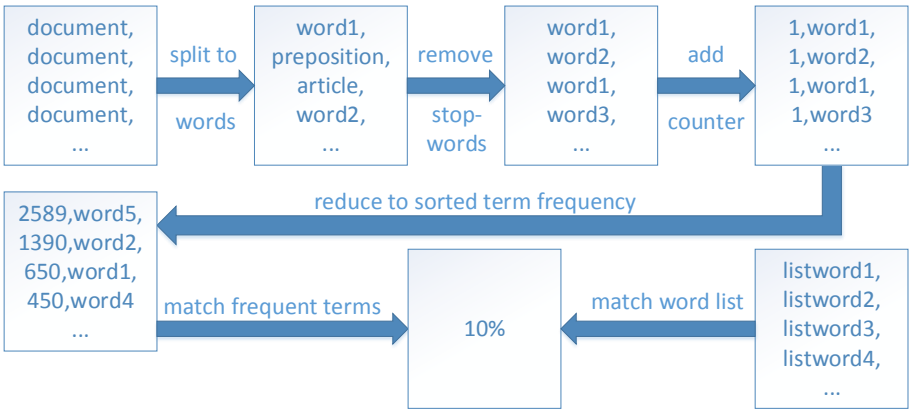

Fig. 5: Workflow of the first instantiation of the framework

4.2 First Results

As proof-of-concept a first test run with a sample corpus of 2315 documents was performed. These documents contain 671,107 words. The word-list of the IEEE Thesaurus contains 45,171 words after cleaning and dividing the grouped terms up into single terms. These terms were searched in the frequent terms of the corpus (1065 terms): 9.77 % (104) of the frequent terms of the corpus were found in the word-list and 0.23 % of the word-list is contained in the corpus. These results are not very surprising, because the corpus and the word-list are not the exactly from the same domain. They both contain scientific content, but the corpus is more in the physics and mathematics domain and the word-list contains more electrical and electronic terms. So the result is as expected. Since the first instantiation showed promising results, a performance test was also performed, for which we describe the results in Section 4.3 below.

The next step involves advanced text analysis methods to build an attributed multiplex graph, based on advanced text mining methods, which recognise the coherencies between the terms. The constructed graph can then have different layers for different relations, also implementing the process as shown in Figure 2.

4.3 Scalability

For the performance test a MapR cluster of 4 machines was used. Each machine is equipped with an Intel i7-6800K with 6 Cores (12 virtual cores), 120 GB SSD for the system and a 1TB SSD for the HDFS cluster. In addition, every node has 64 GB working memory. In total there are 48 virtual cores, 256 GB main memory and 3,5 TB usable in the HDFS. YARN, the resource manager for Spark and other services, has access to 32 cores and 128 GB main memory. The HDFS was configured to work with 4 replica. For Spark the dynamic allocation of the resources is activated. We used Java in version 1.8.0, Spark in version 1.6.1 and Hadoop in version 2.7.0. The tests were performed in three different modes: (1) single node, single core (2) single node, multi core (3) multi node, multi core

The dataset respectively the corpus used for the performance tests contained finally 6000 documents. The procedure started with the processing of 2000 documents; this number was then gradually increased to 6000 documents (in steps of 2000). For each constellation and step the test run was performed five times. The average of the measured runtime (extracted from the Spark history server) is shown in Table 1.

Setup/Number of Documents	**2000**	**4000**	**6000**
Single Node/Single Core	191	451	744
Single Node/Multi Core	37	85	130
Multi Node/Multi Core	66	64	84

Table 1: Runtime of the different setups in seconds

The sample of 2000 documents contains 545,935 words in total. 956 of them occur more than 1000 times in the corpus and 97 of them are also contained in the word list. These are 10,15% of the most frequent terms which are also included in the wordlist. The 4000 documents contain 942,612 words and 1941 most frequent terms. The overlapping words are 171, which makes 8,81% of the most frequent terms. Finally, runs with 6000 documents result in the sum of 1,301,760 of which 2825 belong to the most frequent terms. The number of terms which are in the most frequent terms and the word list are 232. This is 8,21% of the frequent terms.

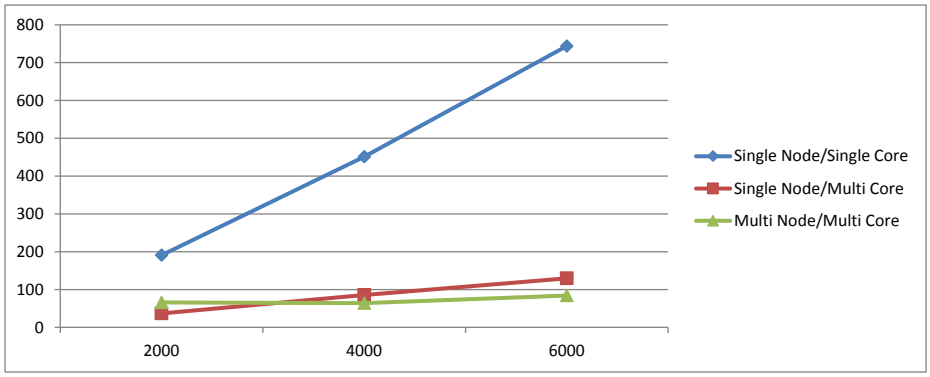

Fig. 6: Visualised runtime of different setups in seconds

The runtimes of the different setups are visualised in Figure 6. On the x–axis the number of documents is shown, and the y–axis displays the time in seconds. So lower points indicate a better performance. Therefore, the line at the top (rhombuses) belongs to the single node, single core setup, which performs the worst. The line with the squares is the single node, multicore constellation which performs best with 2000 documents but is being overtaken by the multi node, multi core setup (line with the triangles) at

4000 documents. This shows that for our framework and lower numbers of documents a single node setup performs better than a cluster of four nodes. The explanation for this is that there is an overhead for distributing and splitting the tasks of application that only pays off if the dataset is big enough. Considering the current setup of the proposed framework regarding the performance aspect the break-even point is reached at around 2600 documents, as seen in Figure 6. However, it must be said here that the integration of further measurement and evaluation methods can impact the derived results.

5 Conclusions

This paper presented a scalable framework for data-driven ontology evaluation, focusing on the scalable processing of large amounts of data such as efficient self-learning support systems [16]. Then, different questions can be tackled, such as how well the ontology fits a given corpus, or which extensions are necessary. We outlined the different tasks of the data-driven evaluation approach, and sketched the modular architecture of our framework enabling a scalable implementation. Furthermore, we discussed first experiments using different instantiations, also focusing on the scalability of the proposed approach. Altogether, our first experiments already show promising results, even with simple instantiations of the framework. As we described in the experimental descriptions, the applied Big Data architecture shows good performance for large corpora demonstrating the scalability of the approach.

Using Big Data techniques and automated processing of large data-sets can deliver a wide range of extensive information, which might lead to a very large set of interim results. This makes it hard to verify manually. In the case of the ontology extension, shown in Figure 4, the number of proposals for modification can however be small enough to be manageable by domain expert, similar to techniques explored in the evaluation of knowledge systems using expert integration, e. g., [23, 21].

It is obvious that for more significant results the frameworks needs more justification. To face this challenge the next steps include a larger number of practically relevant metrics. Therefore, we will use the already computed frequencies of the words and divide them into frequency classes to get the coverage for each frequency class. After that step, the next metric can, for example, include “accuracy” that can be derived by building the average of all frequency classes with more than 60% or a more suitable threshold [22]. Here, the above mentioned metrics will provide a basic set for further explorations.

For future work, we aim at investigating more complex instantiations, in particular focusing on more complex graph- based models including attributed graph structures for ontology evaluation. Here, we will also consider interactive knowledge refinement techniques [1, 2], also building on graph-based description-oriented approaches, e. g., [3]. For the different steps and proposed actions enabled by the framework methods integrating human facets for ontology integration [20], also considering explanation-aware methods [5], seem interesting options to consider for future work. Furthermore, we will target further Big data methods for integration into the proposed framework.

Acknowledgements. The work described in this paper is funded by grant ZIM-KOOP ZF4170601BZ5 by the German Federal Ministry of Economics and Technology (BMWI).

References

1. Atzmueller, M., Baumeister, J., Hemsing, A., Richter, E.J., Puppe, F.: Subgroup Mining for Interactive Knowledge Refinement. In: Proc. Conf. on Artificial Intelligence in Medicine (AIME). pp. 453–462. LNAI 3581, Springer, Heidelberg (2005)
2. Atzmueller, M., Baumeister, J., Puppe, F.: Introspective Subgroup Analysis for Interactive Knowledge Refinement. In: Proc. FLAIRS. pp. 402–407. AAAI, Palo Alto, CA, USA (2006)
3. Atzmueller, M., Doerfel, S., Mitzlaff, F.: Description-Oriented Community Detection using Exhaustive Subgroup Discovery. Information Sciences 329, 965–984 (2016)
4. Atzmueller, M., Kluegl, P., Puppe, F.: Rule-Based Information Extraction for Structured Data Acquisition using TextMarker. In: Proc. LWA. University of Würzburg, Germany (2008)
5. Atzmueller, M., Roth-Berghofer, T.: The Mining and Analysis Continuum of Explaining Uncovered. In: Proc. 30th SGAI Intl. Conference on Artificial Intelligence (AI) (2010)
6. AWS: Amazon web services (2017), https://aws.amazon.com/
7. Biemann, C., Riedl, M.: Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity. Journal of Language Modelling 1(1), 55–95 (2013)
8. Brank, J., Grobelnik, M., Mladenic, D.: A Survey of Ontology Evaluation Techniques. In: Proc. Conference on Data Mining and Data Warehouses (SiKDD 2005). pp. 166–170 (2005)
9. Brewster, C., Alani, H., Dasmahapatra, S., Wilks, Y.: Data driven ontology evaluation. In: International Conference on Language Resources and Evaluation (2004)
10. arXiv Bulk Data Access Amazon S3: test (2017), http://de.arxiv.org/help/bulk_data_s3
11. Bunke, H., Messmer, B.T.: Similarity Measures for Structured Representations. In: Proc. European Workshop on Case-Based Reasoning. pp. 106–118. Springer (1993)
12. Furth, S., Baumeister, J.: TELESUP - Textual Self-Learning Support Systems. In: Proc German Workshop of Knowledge and Experience Management (2014)
13. Hartmann, J., Palma, R., Gómez-Pérez, A.: Ontology repositories. In: Handbook on Ontologies, pp. 551–571. Springer (2009)
14. Hlomani, H.: Multidimensional Data-driven Ontology Evaluation. Ph.D. thesis (2014)
15. Institute of Electrical and Electronics Engineers (IEEE): Thesaurus version 1.0 (2014), http://www.ieee.org/documents/ieee_thesaurus_2013.pdf
16. Knoell, D., Atzmueller, M., Rieder, C., Scherer, K.P.: BISHOP – Big Data Driven Self-Learning Support for High-performance Ontology Population. In: Proc. LWDA 2016. University of Potsdam, Potsdam, Germany (2016)
17. Lee, W.N., Shah, N., Sundlass, K., Musen, M.A.: Comparison of Ontology-Based Semantic-Similarity Measures. In: Proc. AMIA Symposium (2008)
18. Maedche, A., Staab, S.: Measuring Similarity Between Ontologies. In: Proc. 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. pp. 251–263. EKAW '02, Springer-Verlag, London, UK (2002)
19. Mitzlaff, F., Atzmueller, M., Stumme, G., Hotho, A.: Semantics of User Interaction in Social Media. In: Proc. Complex Networks, pp. 13–25. Springer, Heidelberg (2013)
20. Peroni, S., Motta, E., d'Aquin, M.: Identifying Key Concepts in an Ontology, through the Integration of Cognitive Principles with Statistical and Topological Measures. In: Asian Semantic Web Conference. pp. 242–256. Springer (2008)
21. Puppe, F., Atzmueller, M., Buscher, G., Huettig, M., Lührs, H., Buscher, H.P.: Application and Evaluation of a Medical Knowledge-System in Sonography (SonoConsult). In: Proc. 18th European Conference on Artificial Intelligence (ECAI 20008). pp. 683–687 (2008)
22. Spyns, P.: EvaLexon: Assessing Triples Mined from Texts. Technical Report 09, Star Lab, Brussels, Belgium (2005)
23. Supekar, K.: A Peer-Review Approach for Ontology Evaluation. In: 8th International Protege Conference. pp. 77–79. Madrid, Spain (2005)