Utilizing Automatic Predicate-Argument Analysis for Concept Map Mining

Tobias Falke and Iryna Gurevych
Research Training Group AIPHES and UKP Lab
Department of Computer Science, Technische Universität Darmstadt
https://www.aiphes.tu-darmstadt.de

Abstract

Concept maps can be used to provide concise and structured summaries of documents. Motivated by their usefulness in many application scenarios, several approaches have been suggested for concept map mining, the automatic extraction of concept maps from text. However, a major bottleneck of previous work is the common pattern-based approach used to extract concepts and relations from documents which is either limited in coverage or requires a laborious definition of large sets of patterns. Drawing upon recent advances in automatic predicate-argument analysis, we propose to replace pattern-based extraction by using predicate-argument structures. Our experiments compare three different representations with previous work and show that using predicate-argument structures leads to a better extraction performance while being much easier to use.

1 Introduction

A *concept map* is a labeled graph showing *concepts* as nodes and *relations* between them as edges (Novak and Gowin, 1984). They were invented by education researchers in the 1970s and have since been applied in many scenarios, including the usage as a teaching tool (Edwards and Fraser, 1983; Roy, 2008), writing assistance (Villalon, 2012), structure for information repositories (Briggs et al., 2004; Richardson and Fox, 2005) and concise text representation (Valerio et al., 2012). Recently, they have also been proposed as a representation in the context of multi-document summarization and document exploration (Falke and Gurevych, 2017a,b). Their advantages are the concise presentation of central terms and the explicit visualization of relationships. Figure 1 shows an example.

Concept map mining is the automatic generation of concept maps from natural language text. Several approaches have been proposed (see §3). Typically, they extract a set of candidate concepts and relations from given documents and then select a subset of them to construct a concept map. During the extraction, most previous work applies hand-written patterns to extract labels from syntactic representations. As an example, consider the following sentence:

(1) Joseph Novak invented concept maps.

To extract "Joseph Novak" and "concept maps" as concept labels, patterns extracting nsubj- and dobj-dependencies are needed to find the relevant spans when a dependency representation is given. However, these patterns cannot extract anything from the passive variant of the sentence because the relevant tokens now have another grammatical function:

(2) Concept maps were invented by Joseph Novak.

An additional set of patterns would be necessary to handle (2). Due to the syntactic variety of natural language, these pattern-based approaches are either limited in coverage or require a very large and carefully designed set of patterns that covers every possible way in which a proposition can be expressed.

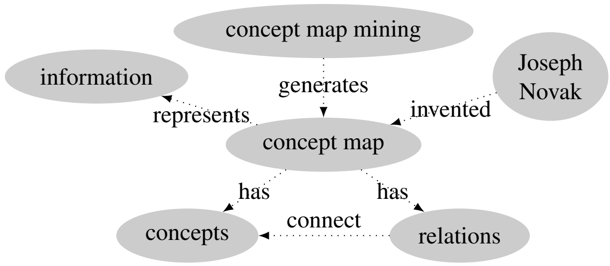


Figure 1: A concept map about concept maps.

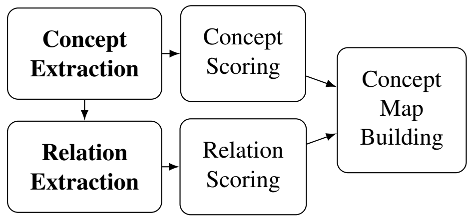


Figure 2: Common concept map mining approach.

To eliminate the high manual effort associated with pattern definition, we propose to utilize semantic instead of syntactic representations as they already abstract away from many syntactic variations. Continuing the example, the binary predicate *invented(Joseph Novak; concept maps)* is a semantic representation for both (1) and (2), requiring no separate handling of the cases. Using such a unified representation based on predicates and arguments, concept map mining approaches no longer need to carefully define large sets of patterns, but can instead make use of existing semantic analysis tools.

In this work, we analyze three different representations that improve upon pure syntactic representations by different degrees: Open Information Extraction (OpenIE) (Banko et al., 2007), identifying predicates and arguments in a sentence, PropS (Stanovsky et al., 2016), which additionally unifies a certain amount of syntactic variations, and Semantic Role Labeling (SRL) (Hajič et al., 2009), mapping predicates and arguments to an abstract, semantic representation.

We describe how these representations can be used and present several experiments comparing them to previous work. As a result, we find that the suggested methods are competitive or even better while requiring no patterns to be defined. This finding has the potential to drastically simplify the development of concept map mining systems in the future, because they can rely on readily available tools for predicate-argument analysis and can focus on subsequent pipeline tasks. In addition, to the best of our knowledge, this is also the first comparative study of concept and relation extraction methods that have been suggested for concept map mining in previous work.

2 Task

Given input documents and a size restriction, extractive *concept map mining* creates a concept map $M = (C, R)$ where every node in C represents a concept, designated by a unique label, and R contains directed edges that connect pairs of concepts with a label such that the resulting proposition describes the concepts' relationship. A concept can be an object, event or abstract idea. All labels for concepts and relations should be taken from the documents and the map has to satisfy the size restriction which defines the maximal number of concepts and relations in the map. Under these constraints, the goal is to select concepts and relations that describe a maximal amount of the document's content. The size restriction controls how much the information should be compressed.

3 Related Work

Several attempts have been made to automatically construct concept maps as defined above (Qasim et al., 2013; Villalon, 2012; Valerio and Leake, 2006, inter alia). Zubrinic et al. (2012) provide a comprehensive survey of work on this task. Figure 2 shows the typical pipeline approach in previous approaches. First, potential labels for concepts and relations are extracted from the documents. Then, they are scored or ranked and a subset is selected and connected to build the final concept map.

In this work, we focus on the extraction part of the pipeline (bold in Figure 2), as this is the step where predicate-argument structures can be leveraged to simplify the syntax-pattern-based strategies of previous work. We briefly present the latter in the following sections.

Concept Extraction The goal of concept extraction is to create a set of potential concept labels, called concept candidates. Ideally, this set is as small as possible while containing all desired phrases. Valerio and Leake (2006) suggest to use constituency parse trees and extract minimal noun phrases from them, i.e. noun phrases that do not cover shorter noun phrases. Both Qasim et al. (2013) and Villalon (2012) work with dependency parse trees and defined patterns to extract phrases from them. While there are slight differences between their patterns, all try to capture single noun tokens, noun compounds and combinations of nouns with adjectives, prepositions and conjunctions. Similar patterns on syntax or part-of-speech sequences were used in other work (Zouaq and Nkambou, 2009; Rajaraman and Tan, 2002; Kowata et al., 2010; Zubrinic et al., 2012).

Relation Extraction During relation extraction, phrases describing the relationship between pairs of concept candidates have to be identified in the text. Following their concept extraction approach, Valerio and Leake (2006) identify verb phrases that connect two concept candidates, whereas Villalon (2012) uses tokens on the shortest path between the candidates in a simplified dependency graph. A pattern-based extraction from dependencies and part-of-speech sequences was also suggested (Qasim et al., 2013; Zouaq and Nkambou, 2009; Rajaraman and Tan, 2002). Olney et al. (2011) use the output of an SRL system and are thus close to this work. However, they focus on a variant of the task where relations are restricted to a fixed domain-specific set of 30 labels, creating less expressive concept maps.

In the next section, we show how the definition of these patterns for both concept and relation extraction can be made obsolete by leveraging existing predicate-argument analysis tools.

4 Concept and Relation Extraction from Predicate-Argument Structures

The intuition for using predicate-argument structures is that a proposition formed by a relation and its concepts in a concept map, e.g. *concept map - has - concepts*, is very similar to a predicate derived by a predicate-argument analysis of a sentence. Therefore, we study how useful different predicate-argument analysis tools are for the extraction of concepts and relations.

Given an input sentence, we want to derive a set P of binary predicates $pred(arg_1, arg_2)$, each of them representing a proposition. We use different existing systems to obtain this set and illustrate their representations for the following example sentence:

(3) Concept maps, which were invented by Joseph Novak, represent concepts and their relationships.

OpenIE The first type of representation we consider is OpenIE. OpenIE systems extract tuples that represent basic binary propositions from a given sentence. Every tuple consists of a relation phrase and two arguments, making the representation very close to concepts and their relations. We use OpenIE-4, a state-of-the-art system.[1] With that system, the extracted tuples P_{OIE} obtained for (3) are:

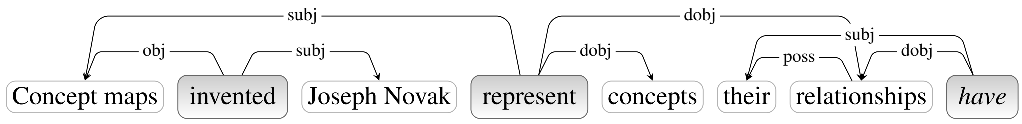
1. were invented (concepts maps; by Joseph Novak)
2. represent (concept maps; concepts and their relationships)

PropS As a second approach, we use PropS, a rule-based converter that turns dependency trees into typed predicate-argument graphs (Stanovsky et al., 2016). In addition to identifying predicates and arguments, it also canonicalizes the representation of propositions, e.g. by unifying variations such as active and passive or copula and appositive constructions. Note that the OpenIE approach does not go that far and only identifies predicate-argument structures.

PropS also classifies predicate-argument relations with a small set of labels such as *subj* and *dobj*. Since it does not try to match any tokens against an inventory of senses or frames, the representation will

[1]`https://github.com/knowitall/openie`, state-of-the-art according to Stanovsky and Dagan (2016b)

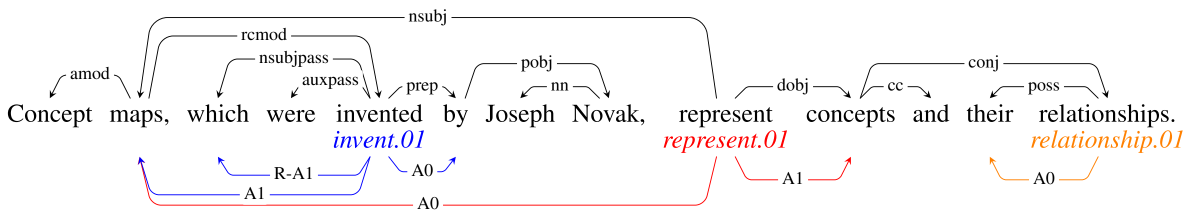
always cover the full content of a given sentence, as opposed to the SRL representation discussed in the next section. For (3), PropS yields the following representation:



To create the set of binary predicates P_{PropS}, we traverse the graph from every predicate node and select as its arguments the subgraphs of the directly connected argument nodes. We remove unary predicates and break down higher-arity predicates by creating all possible pairs except if they have the same edge label (e.g. two objects). Thus, we obtain the following binary predicates for (3):

1. invented (Joseph Novak; concept maps)
2. represent (concept maps; concepts)
3. represent (concept maps; their relationships)

SRL As the third method, we apply semantic role labeling using Mate Tools (Björkelund et al., 2009), which is freely available and was one of the best in the CoNLL 2009 shared task (Hajič et al., 2009). We prefer PropBank-style SRL (Palmer et al., 2005) over FrameNet and VerbNet because of its robustness and maturity. In a sentence, it marks verbs with their PropBank frame, identifies subtrees in the dependency representation of the sentence as arguments and labels them with a role:



This approach is the most sophisticated type of analysis in our study, trying to map a natural language sentence to another layer of abstract semantic representation defined by the frame and role inventory. While being the most powerful representation, it also has disadvantages: First, spans of roles are strictly bound to full subtrees in the dependency parse, as shown in the example, where the relative clause becomes part of the A0-role of "represent". And second, predicates and arguments that are not covered by the inventory of frames and roles cannot be represented.

To obtain the set of binary predicates P_{SRL}, we ignore a predicate completely if it has just one argument, otherwise, we form a binary predicate for every pair of arguments. For (3), this yields the following predicates:

1. invented (concept maps; by Joseph Novak)
2. represent (concept maps which were invented by Joseph Novak; concepts and their relationships)

Given binary predicates P_x derived from a predicate-argument structure $x \in \{OIE, SRL, PropS\}$, concept and relations can be extracted by simply using all arguments of the predicates as the set of concept candidates C_x and the predicates themselves as relation candidates R_x, where each of the latter is associated with two concept candidates as given by the predicate. We intentionally do not further reduce or refine this set with any additional heuristics to assess the out-of-the-box applicability of predicate-argument structures for concept map mining.

Dataset	Gold Concept Maps					Source per Map	
	Maps	Concepts	Length	Relations	Length	Documents	Length
EDUC	30	25.0 ± 0.0	3.2 ± 0.5	25.2 ± 1.3	3.2 ± 0.5	40.5 ± 6.8	97880
BIOLOGY	165	7.0 ± 4.1	1.2 ± 0.5	3.5 ± 3.0	1.9 ± 1.2	1.0 ± 0.0	2621
ACL	230	11.0 ± 5.5	1.8 ± 0.9			1.0 ± 0.0	4987

Table 1: Corpus statistics for the datasets used in experiments. All measures are averages per map and corresponding standard deviations; length values are measured in number of tokens.

5 Datasets

In our experiments, we use corpora of documents paired with gold standard concept maps to evaluate the extraction performance of different approaches. We utilize one recently published dataset and additionally create two other by semi-automatically matching existing maps with corresponding documents.

The first dataset, called EDUC, provides manually created concept maps for clusters of web documents on educational topics. It was created using crowdsourcing and expert annotators and has been recently published by Falke and Gurevych (2017a). In contrast to the other datasets used in this study, this corpus represents a multi-document concept map mining setting, with on average 41 documents and a consequently high number of tokens as the input for a single concept map.

For the second dataset, BIOLOGY, we followed Olney et al. (2011) and used a collection of 464 concept maps produced by experts as teaching materials for biology.[2] The maps were created independently from a text. We matched them automatically with corresponding articles in Wikipedia and manually corrected wrong assignments. Every map is a star-like graph centered around a central concept, e.g. *protein*, and hence has a similar topical focus as an encyclopedic article.

As the third dataset, ACL, we used the ACL RD-TEC 2.0 corpus (QasemiZadeh and Schumann, 2016). It consists of 300 abstracts taken from papers in the ACL Anthology in which two annotators marked terms with a specialized meaning. As abstracts are usually good summaries of a paper, these terms tend to be the central concepts discussed in the papers. We used Apache Tika[3] to extract the full texts, excluding the abstracts, from the PDF version of the corresponding papers. These texts were then used with the annotated concepts as the gold concepts. Note that we cannot use this corpus to evaluate relation extraction, as such annotations are not available.

Table 1 compares the introduced datasets. EDUC represents a multi-document setting, while the other two only have one document per map. However, as this study is focused on the extraction part of concept map mining, this difference is of minor importance. The table also shows that EDUC has the biggest concepts maps with also slightly longer labels for concepts and relations, while the other two datasets, with smaller maps, offer more instances due to the automatic creation approach.

6 Experiments

We conducted several experiments to study the usefulness of the presented predicate-argument analysis tools. In the first two experiments we focus on the recall of different methods during extraction and then turn to the subsequent selection step and corresponding precision evaluations. For all experiments, we preprocessed the documents with components of DKPro Core (Eckart de Castilho and Gurevych, 2014), using tokenization, part-of-speech tagging and constituency parsing from the Stanford NLP tools and Snowball for stemming. Constituency parse trees were converted into collapsed and propagated dependencies (de Marneffe and Manning, 2008).

[2] `http://web.archive.org/web/20120106232123/http://www.biologylessons.sdsu.edu/ta/toc.html` (see *Lesson SemNet* for the different topics)

[3] `https://tika.apache.org/`

Approach	Biology Yield	Biology Recall %	Biology Length	ACL Yield	ACL Recall %	ACL Length	Educ Yield	Educ Recall %	Educ Length
Noun Tokens	51.53	75.61	1.0	48.99	42.48	1.0	167.38	25.07	1.0
Valerio/Leake	69.50	69.60	2.2	77.53	62.21	2.3	406.53	**55.73**	2.4
Qasim et al.	74.61	61.46	2.3	78.04	74.39	2.3	467.15	48.13	2.4
Villalon	60.75	**76.37**	2.3	69.10	**76.09**	2.3	351.85	51.20	2.5
OpenIE	44.53	41.80	4.9	41.99	28.67	5.3	277.70	**58.00**	5.9
SRL	66.28	50.99	4.2	77.21	44.14	5.0	481.59	46.93	6.5
PropS	76.55	**73.50**	3.2	55.87	**58.41**	3.7	451.20	46.27	4.3

Table 2: Concept extraction performance by dataset. For a definition of the metrics, please refer to the text. Bold indicates best recall per group. Concept length given as average number of tokens.

6.1 Concept Extraction

Experimental Setup To evaluate the coverage of different concept extraction strategies, we compared their sets of concept candidates C with the concepts C_G of the gold maps. For every approach, we measured the covered gold concepts with recall $R = |C \cap C_G|/|C_G|$ and used candidate yield $Y = |C|/|C_G|$ to indicate over-generation and thus selection difficulty.[4] Metrics are averages over maps. Concept labels are matched after stemming to allow for morphological variants, e.g. *concept map* and *concept maps*. As a baseline, we applied a strategy that extracts all noun tokens. From previous work, we included the noun phrase strategy of Valerio and Leake (2006) and, as representatives for dependency patterns, the patterns of Qasim et al. (2013) and Villalon (2012) (see §3).

Recall and Yield Table 2 reports results on the three datasets. Out of the different tools used to obtain predicate-argument structures, we observe the highest recall using PropS on two datasets and OpenIE on the other. From previous work, Villalon's method shows the best results on two datasets, while Valerio/Leake's method is best on EDUC. Overall, we conclude that concept extraction based on predicate-argument structure is competitive, giving slightly better (EDUC) or slightly worse (BIOLOGY) results, except for the performance on the ACL datasets. With regard to yield, predicate-argument structures are even more competitive, producing less candidate concepts in most cases. For all approaches, the yield correlates with the size of the input documents, producing most concepts on the EDUC dataset.

One interesting fact is that on EDUC the best performing method, among previous work as well as predicate-argument structures, differs from the one on the other two datasets. The reason is that the concept labels in this corpus tend to be longer, including more complex noun phrases and also verbal phrases describing activities, while concept labels are mostly single nouns in BIOLOGY or noun compounds in ACL. This explains the generally lower recall and better performance of approaches focusing on full noun phrases rather than nouns and noun compounds.

Analyzing the results on the ACL corpus, we found that the automatic extraction of text from the PDFs produced very data, causing a lot of the dependency parses to be of low quality due to wrong sentence segmentation. Interestingly, while this reduced the performance of all approaches using predicate-argument structures, it did not influence the methods from previous work. We hypothesize that these approaches are more robust against these parsing errors because they only extract from dependencies locally, while the other approaches globally process the full parse to derive predicate-argument structures.

Added Concepts We further compared the concept candidate set extracted by the best approach from previous work with those produced by predicate argument structures. To assess whether the latter identified previously uncaptured concepts, we joined both sets and compared the combined recall against the method from previous work alone. In all cases, predicate-argument structures extract at least some concepts that are not covered by previous approaches, with the best approach adding 6.90 (BIOLOGY),

[4]At the current pipeline step, all potential concepts in the text are extracted, while a subset of reasonable size is selected later (Figure 2). Hence, precision and F-scores are rather meaningless; we only report yield. We focus on precision in §6.3.

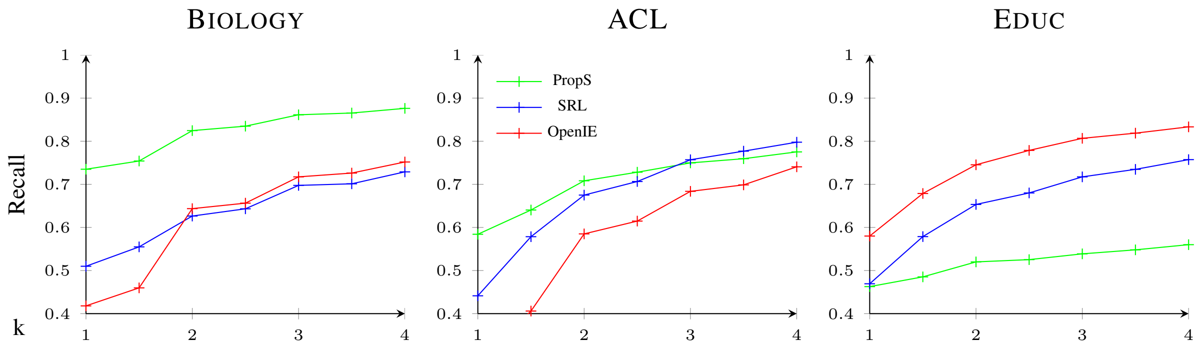


Figure 3: Concept extraction recall for inclusive matches at increasing thresholds k.

5.81 (ACL) and 15.20 (EDUC) points of recall. This shows the predicate-argument structures are not only competitive but can also be used to extend the coverage of previous methods.

Concept Length Finally, we looked at the length of extracted concept labels. As Table 2 shows, the extractions made by previous work tend to be around 2.3 tokens long, while the arguments of predicate-argument structures are up to three times as long. In order to assess whether missing concepts might be present in these longer arguments, we defined a new evaluation metric: An extracted concept c and gold concept c_G match inclusively at k if c_G is contained in c and c is at most $k \cdot |c_G|$ token long. Figure 3 shows the corresponding recall when increasing k. For all three approaches, but especially for SRL and OpenIE, the recall increases dramatically when considering longer arguments. This indicates that the concept extraction performance could be further improved by learning how to reduce longer arguments to the desired parts. Corresponding techniques have been studied, e.g. (Stanovsky and Dagan, 2016a; Stanovsky et al., 2016), however, this step is not trivial as it has to be truth-preserving.

6.2 Relation Extraction

Experimental Setup In our second experiment, we analogously evaluated different relation extraction approaches. To ensure a fair comparison independent of concept extraction, all strategies, including previous work, used gold concepts. Again, we computed candidate recall and yield as our metrics, counting relations that were extracted with a correct label for the correct pair of gold concepts. To account for the fact that some approaches extract complex phrases, e.g. including prepositions, while others extract only single tokens, we used a lenient matching criterion, requiring that the stemmed heads of the relation phrases have to match. Hence, *is located in* and *located* are considered a match. From previous work, we included the verb phrase extraction of Valerio and Leake (2006), the shortest path method of Villalon (2012) and the pattern-based approach of Qasim et al. (2013) (see §3).

Results As shown in Table 3, the shortest path method of Villalon is the best performing method from previous work, however, it also creates a comparably large candidate set. Using predicate-argument structures, we see a substantial improvement on both datasets, and again, PropS performs well on BIOLOGY and OpenIE on EDUC. Note that on both datasets even the other method is on par with Villalon while producing a substantially smaller amount of candidates. We found that the low recall of Valerio/Leake and Qasim's method is due to the small coverage of the patterns, whereas Vilallon's method is very noisy and extracts many meaningless relation phrases. On the other hand, predicate-argument structures benefit from their main advantage in this evaluation: Since all extractions are made on the level of propositions, every concept has at least one meaningful relation to another concept.

With regard to the length of the extracted relation labels (not shown in the table), we found a similar picture: Villalon's method provides very long labels (4.5 token), as it extracts arbitrarily long paths from the dependency structure. SRL yields the shortest labels (1.0), since predicates are restricted to a single

Approach	BIOLOGY Yield	BIOLOGY Recall %	EDUC Yield	EDUC Recall %
Valerio/Leake	2.02	17.75	2.70	8.32
Qasim et al.	1.43	8.08	2.24	3.30
Villalon	9.64	**32.34**	17.28	**21.53**
OpenIE	3.52	31.63	6.47	**25.76**
SRL	4.22	17.57	11.60	17.97
PropS	6.48	**40.95**	11.62	21.20

Table 3: Relation extraction performance.

BIOLOGY Precision@k	ACL Precision@k	EDUC Precision@k
29.46	17.75	**15.07**
24.99	19.47	14.27
32.60	**20.34**	14.80
21.52	10.45	**15.33**
24.11	14.67	13.20
28.86	**16.61**	14.27

Table 4: Concept selection performance (in %).

token by design, while PropS finds a bit longer ones (1.5), including additional auxiliaries and light-verb constructions, and OpenIE extracts the longest labels (2.86), also containing prepositions as in *was made for*. Considering both recall and the style of labels, we found OpenIE to be most useful for this step.

6.3 Concept Selection

Finally, we present a third experiment on concept scoring. While we found predicate-argument structures to be superior for relation extraction in terms of both recall and yield, the picture is less clear for concepts. By analyzing selection performance, we try to shed more light on how useful certain trade-offs between recall and yield are.

Experimental Setup We used the concept candidate sets obtained with the different methods studied in §6.1, assigned a score to each candidate, selected the top-k candidates and compared them to the gold concepts. We set k to the number of gold concepts $|C_G|$ and measured Precision@k. As the score, we use the frequency of the label in the documents, a metric that has been previously proposed to find important concepts (Valerio and Leake, 2006). Note that we are mainly interested in the difference between candidate sets and not the absolute selection performance. Before scoring the candidates, we grouped them by comparing stemmed labels and chose the most frequent label as the representative.

Results Table 4 shows the selection precision of all extraction methods. As expected, the performance closely resembles the picture obtained from the extraction experiment: If the recall is higher after extraction, precision is also higher after the selection. However, an interesting exception is for example PropS and SRL on EDUC: While SRL has a slightly higher extraction recall (46.93 vs. 46.27), PropS selects more relevant concepts (13.20 vs. 14.37), which might be due to the lower extraction yield of PropS.

7 Conclusion

We compared the usefulness of three different approaches for predicate-argument analysis for concept map mining. Comparing them to several previous methods specifically developed for concept map mining, we found that they substantially improve relation extraction while being very competitive with regard to concept extraction. PropS representations are particularly good to capture short noun-focused concepts whereas longer and more complex concepts are more reliably extracted with OpenIE. Considering the good performance and the ease of use – as opposed to manually defining syntactic patterns – future work on concept map mining should rely on ready-to-use predicate-argument analysis for extraction.

Acknowledgments

This work has been supported by the German Research Foundation as part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) under grant No. GRK 1994/1.

References

Banko, M., M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni (2007). Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, Hyderabad, India, pp. 2670–2676.

Björkelund, A., L. Hafdell, and P. Nugues (2009). Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, Boulder, CO, USA, pp. 43–48.

Briggs, G., D. A. Shamma, A. J. Cañas, R. Carff, J. Scargle, and J. D. Novak (2004). Concept Maps Applied to Mars Exploration Public Outreach. In *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain, pp. 109–116.

de Marneffe, M.-C. and C. D. Manning (2008). The Stanford typed dependencies representation. In *Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, United Kingdom, pp. 1–8.

Eckart de Castilho, R. and I. Gurevych (2014). A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, Dublin, Ireland, pp. 1–11.

Edwards, J. and K. Fraser (1983). Concept maps as reflectors of conceptual understanding. *Research in Science Education 13*(1), 19–26.

Falke, T. and I. Gurevych (2017a). Bringing Structure into Summaries: Crowdsourcing a Benchmark Corpus of Concept Maps. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark.

Falke, T. and I. Gurevych (2017b). GraphDocExplore: A Framework for the Experimental Comparison of Graph-based Document Exploration Techniques. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark.

Hajič, J., M. Ciaramita, R. Johansson, D. Kawahara, M. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, and Y. Zhang (2009). The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, Boulder, CO, USA, pp. 1–18.

Kowata, J. H., D. Cury, and M. C. Silva Boeres (2010). Concept Maps Core Elements Candidates Recognition from Text. In *Concept Maps: Making Learning Meaningful. Proceedings of the 4th International Conference on Concept Mapping*, Vina del Mar, Chile, pp. 120–127.

Novak, J. D. and D. B. Gowin (1984). *Learning How to Learn*. Cambridge: Cambridge University Press.

Olney, A., W. Cade, and C. Williams (2011). Generating Concept Map Exercises from Textbooks. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, Portland, OR, USA, pp. 111–119.

Palmer, M., D. Gildea, and P. Kingsbury (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics 31*(1), 71–106.

QasemiZadeh, B. and A.-K. Schumann (2016). The ACL RD-TEC 2.0: A Language Resource for Evaluating Term Extraction and Entity Recognition Methods. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, Portorož, Slovenia, pp. 1862–1868.

Qasim, I., J.-W. Jeong, J.-U. Heu, and D.-H. Lee (2013). Concept map construction from text documents using affinity propagation. *Journal of Information Science 39*(6), 719–736.

Rajaraman, K. and A.-H. Tan (2002). Knowledge discovery from texts: A Concept Frame Graph Approach. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, McLean, VA, USA, pp. 669–671.

Richardson, R. and E. A. Fox (2005). Using concept maps as a cross-language resource discovery tool for large documents in digital libraries. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, Denver, CO, USA, pp. 415.

Roy, D. (2008). Using Concept Maps for Information Conceptualization and Schematization in Technical Reading and Writing Courses: A Case Study for Computer Science Majors in Japan. In *IEEE International Professional Communication Conference (IPCC 2008)*, Montreal, Canada, pp. 1–12.

Stanovsky, G. and I. Dagan (2016a). Annotating and Predicting Non-Restrictive Noun Phrase Modifications. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, pp. 1256–1265.

Stanovsky, G. and I. Dagan (2016b). Creating a Large Benchmark for Open Information Extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, pp. 2300–2305.

Stanovsky, G., I. Dagan, and M. Adler (2016). Specifying and Annotating Reduced Argument Span Via QA-SRL. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, pp. 474–478.

Stanovsky, G., J. Ficler, I. Dagan, and Y. Goldberg (2016). *Getting More Out Of Syntax with PropS*. arXiv:1603.01648.

Valerio, A. and D. B. Leake (2006). Jump-Starting Concept Map Construction with Knowledge Extracted from Documents. In *Proceedings of the 2nd International Conference on Concept Mapping*, San José, Costa Rica, pp. 296–303.

Valerio, A., D. B. Leake, and A. J. Cañas (2012). Using Automatically Generated Concept Maps for Document Understanding: A Human Subjects Experiment. In *Proceedings of the 5th International Conference on Concept Mapping*, Valetta, Malta, pp. 438–445.

Villalon, J. J. (2012). *Automated Generation of Concept Maps to Support Writing*. PhD Thesis, University of Sydney, Australia.

Zouaq, A. and R. Nkambou (2009). Evaluating the Generation of Domain Ontologies in the Knowledge Puzzle Project. *IEEE Transactions on Knowledge and Data Engineering 21*(11), 1559–1572.

Zubrinic, K., D. Kalpic, and M. Milicevic (2012). The automatic creation of concept maps from documents written using morphologically rich languages. *Expert Systems with Applications 39*(16), 12709–12718.