Term Extraction for Domain Modeling

Theresa Kruse [1], Dominic Lohr [2], Marc Berges [2], Michael Kohlhase [1], Halimeh Moghbeli[1], and Marcel Schütz [1]

Abstract: Adaptive learning systems need to use domain and learner models to provide meaningful support for learners. Building fine-grained domain models by hand is very time-consuming, so the demand for partial automation is high. This paper investigates how term extraction tools can support constructing a domain model. Therefore, we study if different automatic term extraction tools give comparable results to a human annotator. Our results show that the current extraction tools support the process, but their results are not directly usable and still need human adjustments.

Keywords: domain modeling, term extraction, adaptive learning system, programming

1 Introduction

Addressing learners' individual needs in large lectures is challenging, especially if we want to tailor teaching materials, learning interventions, and feedback to individual students or sub-cohorts with special needs and educational biographies [DZ11]. AI-based adaptive learning systems (ALS) can help. Given a fine-grained domain model, we work on a system that can maintain a learner model and use it to generate learner-adapted material: targeted explanations, flashcards, or quizzes from collections of learning objects that reference the domain model [Kr23]. In this system, the domain model consists of a knowledge graph of theories, each introducing a set of concepts and their definitions, properties, and relations. The conceptual and semantic relations – i.e., the terminological dependency relation between concepts – in the knowledge graph, together with a model of cognitive processes, shape the learner model. In our experience, given sufficient student activity in the system, the quality and coverage of the domain model are key determinants of the quality of the learner-adaptivity of the system.

The domain model is essential to the system and requires a large investment. Concept definitions and statements of properties and relations can be taken from course material or textbooks. In other words, the development of domain models should be computer-assisted, not only for quality assurance but also for efficiency reasons. In this paper, we present and evaluate experiments on using natural language processing (NLP) techniques, in particular

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg, Wissensrepräsentation und -verarbeitung, Martensstraße 3, 91058 Erlangen, Deutschland, theresa.kruse@fau.de, https://orcid.org/0000-0003-4613-2060; michael.kohlhase@fau.de, https://orcid.org/0000-0002-9859-6337; halimeh.moghbeli@fau.de; marcel.schuetz@fau.de, https://orcid.org/0000-0002-5386-5134

[2] Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstraße 3, 91058 Erlangen, Deutschland, dominic.lohr@fau.de, https://orcid.org/0000-0002-6330-2327; marc.berges@fau.de, https://orcid.org/0000-0002-9982-547X

Gesellschaft für Informatik e.V.: DELFI 2024, pp. 165–182, 2024.
https://doi.org/10.18420/delfi2024_33

term extraction, to generate term lists for domain models. We focus on the following research question: *To what extent can automatic keyword extraction support domain model creation?* To answer this question, we extract terms from an introductory computer science course to compare the results of different automatic term extraction tools with those of manual term extraction.

2 Background

2.1 Domain Modeling for Adaptive Learning Systems

Representing the content of a domain model as a theory graph [Fa18; RK13] allows to keep track of all dependencies automatically and instances of the concepts introduced and presumed in a course offered by an ALS. Together with a model of the competencies of each user, this approach automatically identifies their ability to handle specific concepts. This enables us to provide learners with personalized learning materials that meet their individual needs [AK09; Be23].

2.2 Automatic Term Extraction

For a long time, term extraction has been done based on a combination of linguistic, statistical, and pattern-based methods – combining different metrics based on word frequencies. Thus, the most common words in a terminological text are compared with the most common words in general language. Another metric is term-frequency-inverse document frequency: based on this comparison, values for the termhood of words can be calculated [CP14]. Recently, methods based on large language models (LLM) have become increasingly popular, where word embeddings and cosine similarity are used to estimate termhood values [DMS23].

Regardless of the extraction method, the evaluation measures remain the same, and the quality of a term extraction method in information retrieval is usually evaluated by the precision, recall, and f-measure [Ch92]. To that end, a gold standard is created, usually manually, to serve as a baseline for the evaluation. *Precision* is the percentage of extracted term candidates that are actual terms, while *recall* is the percentage of how many of the terms to be found were extracted. Usually, precision can be improved at the expense of recall and vice versa. The *f-measure* is the harmonic mean between precision and recall and is used to compare systems.

2.3 Term Extraction for Adaptive Learning Systems

Most of the work on term extraction for ALS and domain modeling is based on unsupervised methods because of the large amount of annotated training data required. [Sa19] did a

systematic review of NLP techniques that are used to create concept maps. They show that there is no objective boundary between terms and non-terms when evaluating term extraction tools, and that human experts can disagree on which candidates should be classified as terms. In the following, we present selected work on term extraction for ALS and domain modeling.

[Xu02] follow a frequency-based statistical approach based on GermaNet. Their best precision values vary between 0.52 and 0.63, depending on the corpus size. [BBP13] also use a frequency-based method, which yields a precision of about 0.6 with the best settings. [GN09] take an unsupervised approach based on n-grams. Their results are evaluated against a specialized dictionary. They get an average recall of 0.235 but a precision of 1 with an average f-measure of 0.84. [GA10] extract concepts in mathematics with combined frequency values, also considering how many documents a term candidate appears in. They achieve $f_{0.2}$-measures between 0.67 and 0.94 depending on the document. [ZZY17] use a semantic linked network and get 0.65 for both precision and recall. [BOR23] use different string-matching approaches and achieve the best f-measure of 0.38 with a precision of 0.26 and a recall of 0.36. [So23] extracts terms from PDF textbooks. The texts are split into sentences from which definitions are extracted pattern-based. Similarly, the term candidates are extracted pattern-based from these sentences. The list of terms in the index of the investigated books is used as a gold standard. They get an average recall of 0.647 and a precision of 0.16. To ease the following step of creating a domain model based on the extracted terms, [Ol10] use an approach focused on relations between the concepts that they apply to the domain of biology. Similar work has been done by [Ch21] to create adaptive textbooks and learner models.

3 Methodology

3.1 Extraction Tools

We compare three different term extraction tools that have not been used for the purpose of domain modeling yet: YAKE, D-Terminer, and KeyBERT. All three are directly usable without further training, which is essential for our use case. In building domain models for an ALS, creating training data for each subdomain is not practical.

YAKE uses an unsupervised approach combining different frequency values. Language adaptivity is provided by different stop word lists [Ca20; Ro10]. The scores considered are casing, position, frequency, and distribution. Casing is because uppercase words are more likely to be a term, which is true for English but cannot be directly transferred to German. Regarding position, the authors assume that term candidates from the beginning of a document tend to be more important than those from the end. We need to investigate if that also holds for our text type, namely lecture slides, where concepts introduced at the end of the semester are considered equally important. It is also assumed that more frequent

terms are more important. Regarding the distribution, two aspects are considered: First, the contextuality, i.e., how many different terms a candidate appears within its neighborhood, and second, the distribution over sentences.

D-TERMINER [RHL22; Ri21] is also based on statistical measures, but they are combined with neural network approaches. The data was trained on manually annotated corpora from the domains corruption, equitation, heart failure, and wind energy. Based on traditional frequency-related methods, the extraction is done using conditional random fields and a recurrent neural network with word embeddings. Due to the combination of different domains in the training process, the method should be transferable to new domains.

As a third approach, we apply KEYBERT, which uses the embeddings of the language model Bidirectional Encoder Representation (BERT) [De19; Gr20]. The embeddings are combined with cosine similarities.

3.2 Manual extraction

For our manual extraction, an expert in the domain and its education reads the slides and marks all terms they consider necessary for a domain model. The annotation process is accompanied by several discussions with experts from the domain and from education, as well as knowledge representation to address coding quality issues (i.e., reliability or validity). In these discussions, the coder presents their work and adjusts the process according to the comments, resulting in a final term list.

4 Experiment: Dataset and Results

Our experiment is a case study due to its small scope, focusing on the *Introduction to Programming* as it is part of common undergraduate studies in computer science. We work with a collection of slides from a lecture in German. It contains 892 slides spread over 13 sessions and comprises about 64,300 tokens. The manual annotation yields 1195 terms, out of which 853 are single-word terms (SWT) and 342 multi-word terms (MWT). We further split the annotated MWT into SWT to avoid different criteria being used by the tools to combine single-word terms into multi-word terms. This leaves 1232 terms as a baseline for the automatic extraction.

We apply the three extraction methods from section 2.3 and calculate precision, recall, and f-measure. We extract 2464 SWT with YAKE, i.e., twice as many terms as the expert chose since our goal is a higher recall. Recall is more important than precision in our case because it is less effort for instructors to eliminate false positives (due to a lower precision) than having to think of more concepts for which denominations are missing. YAKE does not lemmatize the words, i.e., it is possible that, for example, one word is extracted twice, in singular and in plural. Still, for the use case of a domain model, we are only interested in

one of these forms, so we regard different forms as a kind of doublet and do a semi-manual lemmatization of the results. Afterward, 1994 term candidates remain. For the extraction with D-Terminer, we use the monolingual extraction with the recommended default settings, i.e., we extract specific terms, common terms, out-of-domain terms, and named entities. We also use the training data from all provided domains (corruption, equitation, heart failure, and wind energy) as none of them is our domain. In D-Terminer, it is impossible to extract only SWT and MWT, nor is it possible to specify the number of term candidates to be extracted. We reduce the extracted list of term candidates by the measures already described for the manual extraction. The extraction yields 1681 term candidates (360 MWT and 1321 SWT); after the reduction, 1425 candidates remain. In KeyBERT, we use the standard configuration to extract 2500 SWT, of which 1921 remain after lemmatization.

	Precision	**Recall**	**F-measure**
YAKE	0.3225	0.5297	0.4009
D-Terminer	0.3818	0.4481	0.4123
KeyBERT	0.2806	0.4523	0.3457

Tab. 1: Results of the automatic extraction compared to the manual work

The precision, recall, and f-measure results are given in Table 1 and are somewhat lower than most of those presented in the related work. We achieved our goal of having recall higher than precision as we extracted way more term candidates than had been manually annotated. We also see that the extraction with KeyBERT gives the lowest f-measure but a recall similar to that of D-Terminer.

5 Discussion

By qualitatively evaluating the results, we find some systems in the false negatives, which are visible throughout the whole dataset, and we do not see systematic mistakes made by one tool or the other. Some errors are due to peculiarities of the German language. For example, the concept *Präzedenz-/Vorrangregel* is introduced by giving two synonymous terms. A human can easily deduce that both, *Präzedenzregel* and *Vorrangregel*, can be used, whereas the automatic extraction can only find *Vorrangregel*. Something similar holds for syntactic variants such as *Name einer Variable* and *Variablenname*, where MWTs are used differently. Another aspect is terms that have a meaning in general language, too, and are thus not recognized as terms, e.g., *faule Auswertung* (Engl. *lazy evaluation*).

We also find a system for classifying false positives. One issue is extra-domain terms in the sense of [Ro20], i.e., words that are terms but not of the domain we are looking at. In our case, these are terms from the domain of academic examinations, e.g., *klausurrelevant* (Engl. *relevant for the exam*), *Tutorin* (Engl. *tutoress*) or *Prüfungsleistung* (Engl. *examination requirement*). This could be avoided by relying on text types other than lecture slides and using only textbooks, although it might take more effort to prepare such data for extraction. Another solution could be domain-specific stop word lists, e.g., one for academic

examination, which can also be reused in other domains. A third aspect is words used in examples without meaning in the domain itself. In our data, a library was used for designing example programs, and thus, words like *Ausleihen* (Engl. *borrow*) or *Bibliotheksmitglied* (Engl. *library member*) were extracted by the tools. It might help to annotate example environments in the data and exclude those parts in the automatic extraction.

The results show a trade-off between the completeness and accuracy of the domain model and the effort required to eliminate unneeded terms. If a high recall is aimed for, as discussed above, the automatically extracted list will contain many terms that need to be removed afterward. On the other hand, if a higher precision is aimed at, concepts will be missing from the list. In our view, adding missing concepts to an extracted list is more time-consuming and error-prone than deleting terms. We believe the precision can be further improved by a stop list adapted to the use case. However, our scores are pretty low, and further research is needed to determine if the tools are a valuable aid for building domain models, i.e., if the time saved by the automatic extraction is worth the errors.

6 Conclusion and Future Work

We presented an experiment to support the creation of domain models with term extraction tools from NLP. It is important to note that the automatic analysis can only consider the terminological level, not the conceptual level. For a full comparison between the automatic and the manual extraction, it would be necessary to create the complete domain model based on both lists and to investigate if the domain model created from the automatically extracted terminology fully covers the concepts from the model based on the manual extraction. Building a complete domain model is tedious, so this comparison is beyond this paper's scope. For these reasons, we cannot finally answer our research question, but we state that automatic extraction could be helpful to ease the creation of domain models. As the manual collection of concepts through extensive reading of the material takes a lot of time, it could be faster if the results of an automatic extraction are used as a baseline to find more terms and build the domain model.

Another area where NLP techniques can help with semantic modeling in ALS is annotating learning objects with domain model URI references for semantic processing. For our system, we have independently developed an LLM-based named entity recognizer for terms and integrated it into the IDE to edit learning objects. This can be seen as another source of coverage requirements, and it would be interesting to combine the two approaches. A practically significant difference between the two approaches is that the requirements from learning object annotation are only generated when the learning model is "used", so it is more of a post-hoc quality evaluation or an incremental way of naturally growing a domain model rather than a "table of contents" from which a domain model can be assembled in parallel as the term lists presented in this paper. We expect the practical change management challenges posed by the two approaches will be quite different.

References

[AK09] Anderson, L. W.; Krathwohl, D. R.: A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. Longman, New York, 2009.

[BBP13] Bordea, G.; Buitelaar, P.; Polajnar, T.: Domain-independent term extraction through domain modelling. In: TIA 2013 - 10th International Conference on Terminology and Artificial Intelligence. Paris, France, 2013, https://hal.science/hal-03826973.

[Be23] Berges, M. et al.: Learning Support Systems Based on Mathematical Knowledge Management. In (Dubois, C.; Kerber, M., eds.): Intelligent Computer Mathematics. Springer Nature Switzerland, Cham, pp. 84–97, 2023.

[BOR23] Banjade, R.; Oli, P.; Rus, V.: Automated Extraction of Domain Models from Textbook Indexes for Developing Intelligent Tutoring Systems. In (Frasson, C.; Mylonas, P.; Troussas, C., eds.): Augmented Intelligence and Intelligent Tutoring Systems. Springer Nature Switzerland, Cham, pp. 124–136, 2023.

[Ca20] Campos, R. et al.: YAKE! Keyword extraction from single documents using multiple local features. Information Sciences 509, pp. 257–289, 2020, https://www.sciencedirect.com/science/article/pii/S0020025519308588.

[Ch21] Chau, H. et al.: Automatic Concept Extraction for Domain and Student Modeling in Adaptive Textbooks. International Journal of Artificial Intelligence in Education 31 (4), pp. 820–846, 2021, https://doi.org/10.1007/s40593-020-00207-1.

[Ch92] Chinchor, N.: MUC-4 Evaluation Metrics. In: Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992. 1992, https://aclanthology.org/M92-1002.

[CP14] Cabre, M. T.; Palatresi, J. V.: 110. Acquisition of terminological data from text: Approaches. In (Gouws, R. H. et al., eds.): Supplementary Volume: Recent Developments with Focus on Electronic and Computational Lexicography. De Gruyter Mouton, Berlin, Boston, pp. 1486–1497, 2014, https://doi.org/10.1515/9783110238136.1486.

[De19] Devlin, J. et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In (Burstein, J.; Doran, C.; Solorio, T., eds.): Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186, 2019, https://aclanthology.org/N19-1423.

[DMS23] Di Nunzio, G. M.; Marchesin, S.; Silvello, G.: A systematic review of Automatic Term Extraction: What happened in 2022? Digital Scholarship in the Humanities 38 (Supplement 1), pp. i41–i47, 2023, eprint: https://academic.oup.com/dsh/article-pdf/38/Supplement_1/i41/50660520/fqad030.pdf, https://doi.org/10.1093/llc/fqad030.

[DZ11] Dosch, M. F.; Zidon, M.: "The Course Fit Us": Differentiated Instruction in the College Classroom. The International Journal of Teaching and Learning in Higher Education 26, pp. 343–357, 2011, https://api.semanticscholar.org/CorpusID:145677236.

[Fa18] Farmer, W. M.: A New Style of Proof for Mathematics Organized as a Network of Axiomatic Theories, 2018, https://arxiv.org/abs/1806.00810.

[GA10] Günel, K.; Aşlıyan, R.: Extracting learning concepts from educational texts in intelligent tutoring systems automatically. Expert Systems with Applications 37 (7), pp. 5017–5022, 2010, https://www.sciencedirect.com/science/article/pii/S0957417409010574.

[GN09] Gunel, K.; Nuriyev, U.: Relation extraction among learning concepts in intelligent tutoring systems. In: 2009 International Conference on Application of Information and Communication Technologies. Pp. 1–5, 2009.

[Gr20] Grootendorst, M.: KeyBERT: Minimal keyword extraction with BERT. Version v0.3.0, 2020, https://doi.org/10.5281/zenodo.4461265.

[Kr23] Kruse, T. et al.: Learning with ALeA: Tailored experiences through annotated course material. In (Klein, M. et al., eds.): INFORMATIK 2023. Designing Futures: Zukünfte gestalten, Berlin. Vol. 337. Lecture Notes in Informatics (LNI) - Proceedings, Köllen Druck+Verlag, Bonn, pp. 395–398, 2023, https://nextcloud.gi.de/s/Y28MkNMjGnE2JY9.

[Ol10] Olney, A. M.: Extraction of Concept Maps from Textbooks for Domain Modeling. In (Aleven, V.; Kay, J.; Mostow, J., eds.): Intelligent Tutoring Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 390–392, 2010.

[RHL22] Rigouts Terryn, A.; Hoste, V.; Lefever, E.: D-Terminer: Online Demo for Monolingual and Bilingual Automatic Term Extraction. In (Costa, R. et al., eds.): Proceedings of the Workshop on Terminology in the 21st century: many faces, many places. European Language Resources Association, Marseille, France, pp. 33–40, 2022, https://aclanthology.org/2022.term-1.7.

[Ri21] Rigouts Terryn, Ayla: D-TERMINE : data-driven term extraction methodologies investigated, eng, PhD thesis, Ghent University, 2021, xxv, 257.

[RK13] Rabe, F.; Kohlhase, M.: A Scalable Module System. Information & Computation 0 (230), pp. 1–54, 2013, https://kwarc.info/frabe/Research/mmt.pdf.

[Ro10] Rose, S. et al.: Automatic Keyword Extraction from Individual Documents. In: Text Mining. John Wiley & Sons, pp. 1–20, 2010, eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470689646.ch1, https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470689646.ch1.

[Ro20] Roelcke, T.: Fachsprachen. Erich Schmidt Verlag, Berlin, 2020.

[Sa19] Santos, V. et al.: Conceptual Map Creation from Natural Language Processing: a Systematic Mapping Study. Revista Brasileira de Informática na Educação 27 (03), pp. 150–176, 2019, http://milanesa.ime.usp.br/rbie/index.php/rbie/article/view/v27n03150176.

[So23] Soliman, A.: An unsupervised linguistic-based model for automatic glossary term extraction from a single PDF textbook. Education and Information Technologies, 2023, https://doi.org/10.1007/s10639-023-11818-1.

[Xu02] Xu, F. et al.: A Domain Adaptive Approach to Automatic Acquisition of Domain Relevant Terms and their Relations with Bootstrapping. In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02). European Language Resources Association (ELRA), Las Palmas, Canary Islands - Spain, 2002, http://www.lrec-conf.org/proceedings/lrec2002/pdf/351.pdf.

[ZZY17] Zhu, M.; Zhao, D.; Yang, J.: A Cognitive-Related Entity and Relation Extraction Model for Online Tutoring Systems. In: 2017 13th International Conference on Semantics, Knowledge and Grids (SKG). Pp. 113–119, 2017.