Automatic concept extraction for learning domain modeling: A weakly supervised approach using contextualized word embeddings

Kordula De Kuthy
Institut für Wissensmedien
Schleichstrasse 6
D-72076 Tübingen
k.dekuthy@iwm-tuebingen.de

Leander Girrbach
Technical University of Munich
Boltzmannstr. 3
D-85748 Garching
leander.girrbach@tum.de

Detmar Meurers
Institut für Wissensmedien
Schleichstrasse 6
D-72076 Tübingen
d.meurers@iwm-tuebingen.de

Abstract

Heterogeneity in student populations poses a challenge in formal education, with adaptive textbooks offering a potential solution by tailoring content based on individual learner models. However, creating domain models for textbooks typically demands significant manual effort. Recent work by Chau et al. (2021) demonstrated automated concept extraction from digital textbooks, but relied on costly domain-specific manual annotations. This paper introduces a novel, scalable method that minimizes manual effort by combining contextualized word embeddings with weakly supervised machine learning. Our approach clusters word embeddings from textbooks and identifies domain-specific concepts using a machine learner trained on concept seeds automatically extracted from Wikipedia. We evaluate this method using 28 economics textbooks, comparing its performance against a tf-idf baseline, a supervised machine learning baseline, the RAKE keyword extraction method, and human domain experts. Results demonstrate that our weakly supervised method effectively balances accuracy with reduced annotation effort, offering a practical solution for automated concept extraction in adaptive learning environments.

1 Introduction

In formal education, the incremental mastery of concepts and the knowledge and competencies that build on them is essential for students to successfully read and understand texts in a specific school subject. Students often struggle to comprehend the relevant concepts within educational materials, leading to difficulties in understanding and applying the knowledge effectively. Many schoolbooks therefore contain a glossary with a list of the key concepts which are manually compiled by the respective schoolbook authors, providing a somewhat subjective list of the relevant concepts.

In the digital counterpart to traditional schoolbooks, computer-based learning platforms, a similar challenge arises: modelling the domain for which the platform provides learning materials and exercises. Most digital systems rely on handcrafted ontologies (or related domain representations) that have been designed by domain experts. They compile a list of domain-specific vocabulary, a very resource-intensive, costly, inefficient, and time-consuming process. In the worst case, such domain ontologies have to be newly constructed for every adaptive learning platform from scratch even if other systems already exist in the same domain. To reduce the effort required, Chau et al. (2021) presented an approach automating concept extraction from digital textbooks. While they demonstrate that such extraction can be successfully carried out, the approach still requires an extensive, domain-specific, manual annotation effort of the textbook as a basis for a supervised machine learning approach. Another particular challenge exists for concept extraction in the educational domain: textbooks not only contain specific vocabulary from one subject domain. In particular, school books usually contain content domain specific words, school domain specific words (homework, teacher, exercise, ...) and example specific words.

Keyword extraction (or concept extraction), a fundamental task in natural language processing (NLP) and information retrieval, aims to identify and extract the most important terms or phrases that best represent the content of a document. These extracted keywords can play a crucial role in various applications, such as document summarization, information retrieval, text classification, and topic modeling, but usually not in the educational domain. Nevertheless, the progress in automatic keyphrase extraction has produced methods that are also useful for the related area of automatic concept extraction from textbooks.

This work focuses on the core task of extracting domain-specific vocabulary. It introduces an approach supported by distributional semantics that

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 175–185

uses contextualized word embeddings, moving beyond simple keyword extraction. Recent methods have explored using word embeddings for concept extraction. However, these methods often have low precision or rely on supervised training with large amounts of labeled data and only use static word embeddings. Some very recent approaches to keyword extraction, such as (Qian et al., 2021), use contextualized word embeddings provided by BERT, which shows improved performance. Nevertheless, these approaches primarily focus on keyword extraction in the scientific domain. This means they aim to extract a few specific keywords from documents that mostly cover one particular topic domain, as noted by (Sammet and Krestel, 2023). These methods still use labeled data and treat contextual word embeddings merely as a more advanced embedding type. This work explores whether the improved performance of contextualized word embeddings also applies to the broader task of glossary extraction in the educational domain. This domain presents a unique challenge due to its multi-theme vocabulary. The approach uses contextualized word embeddings, such as BERT, to select domain-specific expressions in educational texts through a clustering method. Supervision is only required in the form of a small seed list of domain-relevant words. This list can be easily compiled from Wikipedia articles and helps separate clusters of words relevant to the specific domain from those that are specific to the text but belong to a different domain.

Our approach for glossary extraction from structured textbooks could support both glossary building for traditional schoolbooks, domain modeling for adaptive learning platforms, and potentially also student modeling in digital learning environments. Our goal is to create a domain-specific glossary extraction method that accurately reflects the concept annotations made by expert users at the section level. This method can then be used to build both domain and student models for more advanced personalization. To evaluate our method, we assess how closely it matches external expert annotations and internal expert annotations (i.e., glossaries compiled by the schoolbook authors).

2 Related Work

There is a broad number of research strands related to keyword extraction. However, there is little work within the educational field. Therefore, we will focus on approaches with components similar to those in our own method. To present only the main ideas, we will discuss one or two approaches for each method. More detailed overviews are available in (Chau et al., 2021) and (Khan et al., 2022).

Keyword extraction Automatic keyphrase extraction (AKE) has been extensively studied using different approaches, such as rule-based learning, supervised learning, unsupervised learning, or deep neural networks. Since AKE systems are designed to only extract a very small list of relevant keywords, most systems consist of two parts: (1) preprocessing data and extracting a list of candidate keyphrases using lexical patterns and heuristics; and then (2) determining which of these candidates are correct keyphrases. Methods for finding the relevant keyphrases are: statistical methods or frequency-based methods, clustering-based methods, graph-based methods, embedding-based methods, and machine learning methods.

The most basic frequency-based approach is the statistical measure tf-idf (term frequency-inverse document frequency (Jones, 2004)). This method effectively finds relevant terms within a document (high recall) but often includes many irrelevant terms (low precision). Therefore, most approaches combine tf-idf with other measures to narrow down the list of potential keywords.

In graph-based approaches, an entire document is modeled as a graph of semantic relationships between the terms and a ranking approach then selects the terms with the highest number of relationships. Prominent approaches are (i) RAKE (Rose et al., 2010) in which a graph of word co-occurrences is constructed and the top ranked words in this graph are extracted as key words, (ii) TextRank (Mihalcea and Tarau, 2004) in which documents are represented as undirected and unweighted graphs and (iii) PositionRank (Florescu and Caragea, 2017), a fully unsupervised, graph-based model, that simultaneously incorporates the position of words and their frequency in a document to compute a PageRank score for each candidate word. The most recent graph-based approaches employ contextualized word embeddings for calculating the ranking, cf. KPRank (Patel and Caragea, 2021).

In clustering-based approaches, clustering algorithms group candidate phrases into topic clusters and the most representative ones from each cluster are selected as key phrases. Liu et al. (2009) employ cooccurrence-based term relatedness, and

a Wikipedia-based term relatedness for clustering. Grineva et al. (2009) develop a graph-based approach for identifying domain specific terms in multi-theme documents - an unsupervised topic-based clustering method that partitions a graph into thematically cohesive groups of terms.

In supervised statistical learning approaches, all terms in a document must be classified as either positive or negative instances of relevant keyphrases. This classification is based on patterns learned from annotated training sets. For example, Hulth (2003) define manual rules combined with frequency measures to extract all potential keyword expressions from a text. A classifier then determines which of these are actual keyword expressions. Current methods use word embeddings to represent words. For instance, Wang et al. (2014) examine word embeddings to measure the relationships between words in graph-based models. Recent methods also use neural networks (cf. Zhang et al., 2016).

In approaches that view AKE as a sequence labelling task, Alzaidy et al. (2019) predict a sequence of labels where the two labels are keyphrase word or non-keyphrase word. The recent availability of contextualized word embeddings has enabled further improvement in AKE as sequence labelling, as in (Sahrawat et al., 2019) or (Sammet and Krestel, 2023) where a fine-tuned BERT labels relevant keyphrases in abstracts from economics articles.

Concept extraction In concept or term extraction approaches, the goal is to extract not only a small list of the most general candidates but also extract more specific terms that can be used in applications such as domain ontology construction, text classification, or information extraction. The two possible approaches here are constructing a domain model from scratch or using contrastive corpora to identify domain-relevant terms.

Bordea et al. (2013) propose a domain-independent method for extracting terms. They find general terms in a document, similar to keyphrase extraction, and then use these to build a domain model. Based on this model, they identify other semantically similar terms in the document. The method's performance varies across domains but is more stable than basic term extraction approaches like TermExtractor.

Only a few methods address concept extraction in education. One method, proposed by Chau et al. (2021), uses a supervised feature-based machine learning approach to automatically extract concepts from digital textbooks. This method trains a supervised learning model to classify whether a term or phrase is a concept. It bases this classification on a detailed set of features. One of the few approaches that explicitly aims at constructing domain-specific glossaries, presented by Park et al. (2002), focuses on building domain-specific glossaries. This is similar to the goal of this article. This method uses a tf-idf-based approach.

Ontology extraction Textbooks and the educational domain play a greater role in the domain of ontology extraction, i.e, building concept hierarchies for textbooks or ontologies from textbooks.

(Wang et al., 2015) present an approach that uses Wikipedia as an external resource to build a concept hierarchy for textbooks. The goal is to extract keyphrases for each chapter of a given book. First, they extract a set of related and important Wikipedia concepts for each book chapter. Second, they use local features to extract related concepts for each chapter separately, utilizing measures such as textual similarity between a book chapter and candidate concepts. The resulting candidate set consists of the top N candidates based on their cosine similarity score and those candidates whose title appears in the chapter title (i.e., `titleMatch` equals 1). These two simple but powerful features can capture most of the related and important concepts for each book chapter.

A similar approach is described in (Conde et al., 2016). This paper introduces LiTeWi, a method that combines term extraction techniques (like linguistic filters and tf-idf) with Wikipedia. It uses Wikipedia as a knowledge base to improve term extraction accuracy by removing terms not related to Wikipedia entries within the specified domain.

Summing up, to the best of our knowledge, current automatic term and concept extraction methods perform unexpectedly poorly and are not tailored for the educational field. Improving automatic extraction of domain-specific concepts would be beneficial for immediate tasks such as student modeling and content recommendation in learning platforms or tutoring systems. Furthermore, it would advance the automatic extraction of domain, i.e. specific glossaries and the construction of ontologies, both of which are crucial for developing learning platforms that currently rely heavily on manual domain models.

For documents with multiple themes, clustering seems to be the most promising approach. This

method has been mainly used for extracting keywords. To encode the domain-specific meaning of concepts that require clustering, contextualized word embeddings seem to be the most promising approach. However, these embeddings have only been used for supervised single-word or sequence labeling of keywords in scientific documents. Our work combines these two methods for domain-specific vocabulary extraction. Our method outperforms other methods and does not require large amounts of labeled data for training and testing.

3 Method

In our approach, called GlossEx, we extract concepts specific to a given domain from text. We do not just extract a small list of keyphrases. Instead, we extract all phrases or words that represent the main concepts of that text. This creates a specialized vocabulary list, which is similar to manually compiling a glossary for a specific text.

3.1 Task formulation and dataset

We are trying to solve the following technical task: Given a document $\mathcal{D}$ that represents a specific domain, our goal is to extract the specialized vocabulary $\mathcal{V}$ of that domain from $\mathcal{D}$. We are exploring this task within the domain of teaching economics in schools. For our dataset, we selected 28 economics textbooks used for the economic curriculum in German secondary schools. We expect our method to identify domain-specific concepts such as "workforce", "consumption", "entrepreneur", and similar terms. In order to extract the domain-specific vocabulary, we propose the following pipeline:

1. Document preprocessing, i.e. tokenization, lemmatization, POS-tagging, ...
2. Extract salient vocabulary $\mathcal{S}$ contained in $\mathcal{D}$
3. Cluster vocabulary items in $\mathcal{S}$ based on their contextualised embeddings
4. Obtain $\mathcal{V}$ by filtering $\mathcal{S}$ using limited domain knowledge

This pipeline is based on the following observations: Because $\mathcal{D}$ represents a specific domain, it features specialized vocabulary. Conversely, this specialized vocabulary is particularly prominent in $\mathcal{D}$ compared to general, non-domain-specific documents. The second step of the pipeline uses this observation. However, economic textbooks contain three distinct types of salient vocabulary in addition to the general vocabulary found in any text: (i) specialized vocabulary (which is the extraction target), (ii) education-specific vocabulary (such as instructions like "write" or "analyze"), and (iii) example vocabulary (which appears prominently due to its presence in running or repeated examples).

Therefore, we need to exclude education specific vocabulary and example vocabulary from $\mathcal{S}$ in order to obtain $\mathcal{V}$. This is done through Items 3 to 4. The clustering step in Item 3 serves to stabilise the filtering method in Item 4: We observe that contextual embeddings form useful clusters, so that specialized and non-specialized vocabulary form local clusters in embedding space. Therefore, we exploit this property to include or exclude complete clusters in $\mathcal{V}$ instead of single lemmas. Item 4 accesses limited domain knowledge to differentiate between the 3 salient categories described above. We use the limited domain knowledge to label each cluster with one of the three categories listed above, and eventually only return lemmas in clusters labeled as specialised vocabulary.

Next, we describe in detail how to implement each step of the proposed pipeline. The focus is on German economics textbooks, but the general method applies to various domains and languages, provided the necessary models are available. We also present the specific German processing tools.

3.2 Preprocessing

The NLTK library (Bird et al., 2009) is used for splitting sentences and tokenizing text. The Hanover Tagger (Wartena, 2019), which is specifically designed for German, is used for sentence-level lemmatization and POS tagging. All subsequent steps are applied to the lemmatized document $\mathcal{D}$, unless stated otherwise.

3.3 Extracting Salient Vocabulary

We extract the salient vocabulary $\mathcal{S}$ from $\mathcal{D}$ using the method proposed by Lemay et al. (2005). This method calculates scores for all lemmas. These scores show whether a lemma appears more often in $\mathcal{D}$ than is typical for the language in general. Thus, this method distinguishes the salient vocabulary of $\mathcal{D}$ from general vocabulary. For evaluation, we use two general German word frequency lists:

1. A frequency list[1] derived from the DeReKo

[1] `DeReKo-2014-II-MainArchive-STT.100000` obtained

(Lüngen, 2017). DeReKo is a very large corpus that is representative of contemporary German.

2. The SUBTLEX-DE frequency list (Brysbaert et al., 2011), which has been shown to better explain cognitive saliency of words in decision time experiments.

We only consider nouns and verbs, and we discard stopwords and lemmas that appear less than four times in $\mathcal{D}$, as well as tokens that contain special characters. Note, that the method described in (Lemay et al., 2005) differs from tf-idf. Specifically, tf-idf calculates frequencies only within a single corpus, whereas our method compares frequencies between two corpora.

3.4 Clustering Vocabulary

We cluster lemmas in $\mathcal{S}$ by agglomerative clustering of contextualised embeddings. To compute embeddings, we use the `bert-base-german-cased` BERT model provided by Chan et al. (2020). We embed each (non-lemmatized) sentence individually (after subword tokenization). Then, embeddings of subword tokens are mean-pooled to derive embeddings of the original tokens. Finally, lemma embeddings are the mean of all token embeddings associated with the respective lemma.

Agglomerative clustering is computed by the respective scikit-learn implementation (Pedregosa et al., 2011) using default parameters. In preliminary experiments, we found agglomerative clustering to perform better for our task than k-means clustering or spectral clustering methods. We set the number of clusters (which is a required parameter of agglomerative clustering) to $\frac{|\mathcal{S}|}{4}$. This means the expected number of words in a cluster is 4.

This approach differs from graph-based algorithms, such as the one proposed by Grineva et al. (2009). We do not use graph topology to find clusters. Instead, we directly cluster lemmas in the embedding space. In the graph paradigm, this means we are working with a fully connected graph where edge weights are determined by a distance metric in the embedding space.

3.5 Filtering by Domain Knowledge

In the last step, we select clusters that contain specialized vocabulary from a specific domain. However, obtaining this information directly from embeddings is difficult. Therefore, we create two lists: $\mathcal{V}_{\text{edu}}$ and $\mathcal{V}_{\text{eco}}$. $\mathcal{V}_{\text{edu}}$ contains seed words related to the education domain, and $\mathcal{V}_{\text{eco}}$ contains seed words related to the economics domain. These lists inject a limited amount of domain knowledge into our method, which helps us determine if a cluster contains terms associated with the education domain, the economics domain, or neither.

Application of seed lists Each cluster $\mathcal{C}$ (representing a set of lemmas in $\mathcal{D}$) receives two scores: an association score for educational vocabulary (σ_{edu}) and an association score for economics vocabulary (σ_{eco}). The scores for a cluster are calculated by taking the average of the 10 smallest pairwise distances between any word in that cluster and any word in either the educational vocabulary ($\mathcal{V}_{\text{edu}}$) or the economics vocabulary ($\mathcal{V}_{\text{eco}}$). The distances between words are measured using the Euclidean distances of fastText embeddings[2] (Grave et al., 2018). It is important to note that this method uses static word embeddings, which differs from the approach in Section 3.4 where contextualized embeddings from a German BERT model (Chan et al., 2020) are used. fastText embeddings are chosen because their model can create embeddings for any string based on its character n-grams. This avoids the problem of out-of-vocabulary words. Specifically, clusters are kept if they meet one of the following conditions:

$$\sigma_{\text{eco}} + 0.03 < \sigma_{\text{edu}} \quad (1)$$

$$\sigma_{\text{eco}} < \min\{0.3, \sigma_{\text{edu}}\} \quad (2)$$

In simpler terms, this means that clusters are selected if they are generally close to the economics vocabulary ($\mathcal{V}_{\text{eco}}$) or if they are significantly closer to $\mathcal{V}_{\text{eco}}$ than to the educational vocabulary ($\mathcal{V}_{\text{edu}}$). These thresholds are specific to the embedding space used and are set manually. The thresholds were determined before any labeled data was available, so their manual setting does not affect the validity of the results. With a small amount of labeled data, it would be possible to automatically adjust these thresholds.

Construction of seed lists The lists are created independently from the evaluation data to avoid circularity. With the PetScan interface we extracted Wikipedia article titles and wikidata entity names

from `https://www.ids-mannheim.de/digspra/kl/projekte/methoden/derewo/`

[2]obtained from `https://fasttext.cc/docs/en/crawl-vectors.html`

with the following hyperparameters: To populate $\mathcal{V}_{\text{edu}}$, we run one query on the "Bildung" (engl.: *education*) category with maximum depth 6 and require the found pages to link to the Wikipedia page "Schule" (engl.: *school*). To populate $\mathcal{V}_{\text{eco}}$, we run two queries on the "Wirtschaftswissenschaft" (engl.: *economics*) category with maximum depth 6. For the first query, we require found pages to link to the Wikipedia page "Markt" (engl.: *market*). For the second query, we require found pages to link to the Wikipedia page "Bedarf" or to the Wikipedia page "Bedürfnis" (both engl.: *need*). We combine the results of both queries.

To create the final seed lists, which contain only single lemmas, the preprocessing method described in Section 3.2 is applied to every page title returned by PetScan. The resulting lemmas are then saved. Consequently, $\mathcal{V}_{\text{edu}}$ contains 562 unique lemmas, and $\mathcal{V}_{\text{eco}}$ contains 677 unique lemmas. Although these lists may seem large, they contain a significant amount of noise. Additionally, as shown in Section 4.3, extracting specialized vocabulary using only the words in the seed lists, without our GlossEx method, leads to poor performance.

4 Results and Evaluation

The main evaluation metrics are precision and recall. The goal is to assess how much of the specialized vocabulary the proposed method finds and how many lemmas it returns are actually specialized vocabulary.

4.1 Data

To evaluate the GlossEx method, we use 28 partially digitized German economics textbooks, which cover various school types and years. Nineteen of these textbooks also have paired OCR-scanned glossaries. For all lemmas that appear at least four times in the corpus, we collect expert judgments whether each lemma is domain-specific vocabulary in the field of economics. One expert (not an author of this paper) labeled all 3,458 unique lemmas with binary labels. Out of these, 469 lemmas (13.56%) are labeled as domain-specific vocabulary. To assess inter-annotator agreement, another expert (also not an author of this paper) independently labeled a subset of 510 lemmas. Cohen's κ (Cohen, 1960) between both annotators is 0.66, which is considered substantial agreement according to Landis and Koch (1977). Furthermore, the f1-score between both annotators is 0.79, which sets an upper bound on the models' performance. However, this bound is not specific to any particular textbook.

4.2 Baselines

We compare our method GlossEx described in Section 3 to several baselines. The first set of baselines comprises methods that are widely used for keyword extraction, namely tf-idf (Jones, 2004), Rapid Automatic Keyword Extraction (RAKE) (Rose et al., 2010), and supervised learning on static word embeddings. Note, that the supervised baseline requires labels and therefore uses strictly more information than is available to our method. Thus, the supervised baseline serves as an upper bound to see how much worse methods that do not require explicit labels perform.

A second set of baselines evaluates how well we can extract keywords using two methods: either by simply using the seed lists from Section 3.5 or by using the glossaries included in textbooks. By comparing two seed lists as a baseline, we ensure that our algorithm can discover domain-specific vocabulary beyond the initial input. By comparing two glossaries as a baseline, we confirm the relevance of our problem. This comparison shows that textbook glossaries do not contain all the vocabulary that experts consider domain-specific.

tf-idf measures how relevant a term is to a specific document within a collection of documents (corpus). A term is more relevant if it appears often in the document (high term frequency) but less relevant if it appears in many other documents (high inverse document frequency). To apply tf-idf in our context, given a textbook $\mathcal{D}$: Term frequency is the frequency f_t of term t in $\mathcal{D}$. Inverse document frequency is the logarithm of the inverse ratio of sections in $\mathcal{D}$ that also contain t. The formula for tf-idf is:

$$\text{tf-idf}(t, \mathcal{D}) = \frac{f_t}{\sum_{t'} f_{t'}} \cdot \log\left(\frac{N}{g_t}\right) \qquad (3)$$

Here, N is the total number of sections in $\mathcal{D}$, and g_t is the number of sections in $\mathcal{D}$ that contain the term t. Typically, a threshold $\tau \in \mathbb{R}$ is used to identify domain-specific vocabulary. Any term with a tf-idf score greater than τ is considered part of this vocabulary. In our specific case, we choose the τ value that maximizes the f1-score of the predicted domain-specific vocabulary.

Rapid Automatic Keyword Extraction RAKE (Rose et al., 2010) selects keyphrases from documents for information retrieval via assigning each keyphrase a score based on cooccurrence statistics and returning the 33% top scoring keyphrases. We use the implementation provided by the `rake-nltk` library.[3] We only consider single keywords, i.e. the maximum keyphrase length is 1. As stopwords, we provide the list of German stopwords provided by the `NLTK` (Bird et al., 2009).

Supervised Learning The task of domain-specific vocabulary extraction can be described as a binary classification problem if we are given lemmas and binary labels that show if each lemma is specific to a certain domain. We represent lemmas using their static fastText embeddings, as described in Section 3.5. Then, we train a multi-layer perceptron (MLP) to predict the correct label from these embeddings. To get predictions for all lemmas in a textbook, we use 5-fold stratified cross-validation. We use the scikit-learn library for both cross-validation and the MLP.

Glossaries Nineteen textbooks in our dataset include a glossary. We assess how much of the domain-specific vocabulary these glossaries cover and whether they also contain general, non-domain-specific vocabulary. We extract all elements from these 19 glossaries. We then keep all individual tokens, excluding stopwords, and lemmatize them. From these, we only keep nouns and verbs. We then return only the remaining lemmas from the glossary that appear in the given textbook $\mathcal{D}$.

Seed Lists In this case, we return all entries from the seed lists described in Section 3.5 that also appear as domain-specific vocabulary in the textbook. This baseline tests whether GlossEx can discover new domain-specific vocabulary and successfully discard non-domain-specific vocabulary. However, the seed lists directly determine which lemmas are returned as domain-specific and which are discarded after the clustering step (see Section 3.4). Therefore, there is a close relationship between the precision of the seed lists (i.e., how many of the seed list entries are actually domain-specific vocabulary) and the precision of GlossEx.

4.3 Results

Overall Performance In Table 1, we present the precision, recall, and f1-score for all methods. These scores are macro-averaged across all 28 textbooks in the dataset. The supervised baseline shows the best overall performance, as expected. Because the data is imbalanced (with only a few domain-specific words), the precision for this baseline is higher than its recall. Conversely, tf-idf and RAKE perform poorly in terms of f1-score. These methods identify many words as domain-specific vocabulary, leading to high recall but low precision. RAKE performs better than tf-idf, even though the optimal score threshold is used for tf-idf.

Method	Precis.	Recall	F1
tf-idf	0.152	<u>0.685</u>	0.230
RAKE	0.172	**0.854**	0.283
Glossary	**0.821**	0.258	0.382
Wiki-Seedlist	0.367	0.065	0.103
GlossEx-dereko (ours)	<u>0.543</u>	0.584	<u>0.545</u>
GlossEx-subtlex (ours)	0.518	0.645	**0.559**
Supervised	0.754	0.524	0.589

Table 1: Precision, recall, and f1-scores of GlossEx and baselines. Scores are averages across the 28 textbooks in our dataset. Best results (excluding supervised) are in bold, and second best results are underlined. "dereko" and "subtlex" refer to the background corpus.

Using glossaries improves performance. However, these results cannot be directly compared because glossaries are only available for 19 textbooks. Therefore, the reported results are averaged only over these 19 textbooks. Generally, glossaries mainly contain domain-specific vocabulary. However, they miss 75% of the domain-specific vocabulary in textbooks, which is indicated by the low recall. The seed list extracted from Wikipedia yields low precision, low recall, and consequently, a very low f1-score. Still, the precision is higher than that of tf-idf and RAKE. This is expected because the construction method directly uses the Wikipedia category hierarchy. This primarily confirms that our method's performance is not simply due to a very strong starting point through the seed lists.

Finally, GlossEx achieves an improvement over the seed lists in terms of f1-score and recall. This shows that GlossEx can indeed leverage distributional semantics to identify domain-specific vocabulary. Our method also significantly reduces the gap between baseline methods and supervised learning. Compared to supervised learning, our method achieves higher recall at the expense of

[3] https://pypi.org/project/rake-nltk/

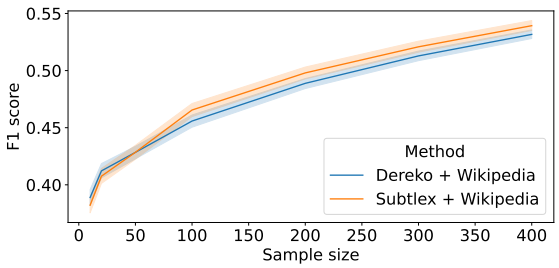

Figure 1: GlossEx f1-score vs. increasing seed list size.

lower precision. Therefore, exploring more precise methods to characterize the contextual semantics of words in textbooks seems to be a promising direction for improving our method.

Since GlossEx relies on external data, we must characterize the influence of the background corpus and seed list on its performance. As seen in Table 1, using SUBTLEX-DE instead of DeReKo (referred to as "subtlex" and "dereko") as the background corpus results in higher recall but lower precision. One possible reason for this is that DeReKo has broader vocabulary coverage, which leads to fewer lemmas appearing prominent for a given document. Additionally, the DeReKo corpus contains many newspaper texts, which might bias frequency estimates for terms related to topics like politics and financial news.

Effect of Seed List Size To assess how the size of the seed list affects our method's performance, we repeatedly select random seed lists of different sizes. These sample sizes, denoted as n, are chosen from the set $\{10, 20, 50, 100, 200, 300, 400\}$. From the complete set of all entries in the seed lists, for each sample size n, we sample $k = 100$ seed lists. In all instances, both the economics and education seed lists have the same number of entries. We then re-evaluate our method using these selected seed lists.

Figure 1 shows that the performance of GlossEx consistently improves as the seed list size increases. This outcome is expected and these findings also indicate that GlossEx is resilient to direct overlaps between seed lists and textbook vocabulary. The Wikipedia seed list, for example, contains only a few domain-specific terms. However, GlossEx can fully utilize the semantic information found in these entries. In summary, our results demonstrate that GlossEx performs well with 100 to 200 noisy seed words. However, it achieves optimal performance when provided with more, higher-quality entries.

5 Discussion and Future Work

Our method, GlossEx, uses traditional machine learning and natural language processing (NLP) techniques for domain vocabulary extraction, such as clustering and word embeddings. Unlike previous methods, we also include contextualized embeddings derived from large language models (LLMs). Recent versions of generative LLMs have been very successful in various zero-shot applications (Brown et al., 2020; Achiam et al., 2023). These advancements are promising for all areas of NLP, including education (Alhafni et al., 2024; Wen et al., 2024), making the use of LLMs for domain-specific vocabulary extraction in a zero- or few-shot manner an exciting direction for future research. However, we believe that combining LLMs with modular approaches like ours is most effective, because we can not only identify, but also explain *why* certain words are considered domain-specific. This explanation comes from traceable differences in word occurrences in domain-specific versus general texts, and from semantic similarity to known domain-specific words. This built-in interpretability makes GlossEx a valuable approach even in the era of LLMs.

6 Conclusion

Given educational materials, how can we systematically extract the domain concepts to be learned and understood by students? Answering this is relevant for building glossaries for textbooks, for domain and student modeling for adaptive learning platforms, and for the automatic derivation of activity models for text-based learning materials. In this paper, we investigated how computational linguistic methods such as distributional semantic analysis and clustering can be combined to automatically extract a domain-specific glossary. We presented a pipeline to extract specialized vocabulary from single documents, e.g. , textbooks. The pipeline is optimized for documents from the educational domain, where pedagogical terminology cannot easily be separated from subject domain concepts by statistical methods alone. Pursuing a weakly supervised approach, we injected only a limited amount of domain knowledge in the form of a seed list readily obtained from Wikipedia. We evaluated the method on German economics textbooks. Evaluation is both automatic, by comparing the extracted vocabulary to paired glossaries, and manual by human domain experts.

Data and Code Availability

Our implementation of GlossEx is available at https://github.com/LGirrbach/GlossEx. We cannot release the textbook material used in this paper because it is copyrighted. However, the textbook titles are included in our code release.

Limitations

While our approach to automatic concept extraction using contextualized word embeddings and weakly supervised learning shows promising results, there are some limitations to our approach.

First, the reliance on pre-trained language models such as BERT, which are primarily trained on general corpora, may not fully capture the nuances of domain-specific language used in educational texts. This can lead to less optimal performance in identifying and clustering domain-specific vocabulary, particularly in specialized fields not well-represented in the training data.

Second, the quality and comprehensiveness of the seed lists used to guide the clustering process significantly influence the results. Although we used Wikipedia to generate these lists, the potential gaps in coverage can affect the accuracy of the extracted concepts. In future work, one could explore more refined methods for seed list generation or incorporate additional domain-specific resources to support the robustness of the approach.

Third, the performance of GlossEx is evaluated on a relatively small and specific dataset of German economics textbooks. This limits the generalizability of our findings to other subjects and educational contexts. Extensive testing on diverse datasets is necessary to validate the broader applicability of our approach.

Finally, while our approach reduces the need for extensive manual annotation, it still requires some level of domain knowledge for seed list creation and cluster validation. This semi-supervised nature means that the method is not entirely free from human intervention, which could be a limitation in fully automating the concept extraction process.

Addressing these limitations in future research will be crucial for enhancing the scalability, accuracy, and applicability of our method in various educational settings.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. In *arXiv*.

Bashar Alhafni, Sowmya Vajjala, Stefano Bannò, Kaushal Kumar Maurya, and Ekaterina Kochmar. 2024. Llms in education: Novel perspectives, challenges, and opportunities. In *arXiv*.

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The World Wide Web Conference*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with python: analyzing text with the natural language toolkit.

Georgeta Bordea, Paul Buitelaar, and Tamara Polajnar. 2013. Domain-independent term extraction through domain modelling. In *International Conference on Terminology and Artificial Intelligence*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *NeurIPS*.

Marc Brysbaert, Matthias Buchmeier, Markus Conrad, Arthur M Jacobs, Jens Bölte, and Andrea Böhl. 2011. The word frequency effect. In *Experimental psychology*.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *COLING*.

Hung Chau, Igor Labutov, Khushboo Thaker, Daqing He, and Peter Brusilovsky. 2021. Automatic concept extraction for domain and student modeling in adaptive textbooks. In *International Journal of Artificial Intelligence in Education*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. In *Educational and psychological measurement*.

Angel Conde, Mikel Larrañaga, Ana Arruarte, Jon A Elorriaga, and Dan Roth. 2016. litewi: A combined term extraction and entity linking method for eliciting educational ontologies from textbooks. In *Journal of the Association for Information Science and Technology*.

Corina Florescu and Cornelia Caragea. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *ACL*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *LREC*.

Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In *18th international conference on World wide web*.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *EMNLP*.

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. In *Journal of documentation*.

Muhammad Qasim Khan, Abdul Shahid, M Irfan Uddin, Muhammad Roman, Abdullah Alharbi, Wael Alosaimi, Jameel Almalki, and Saeed M Alshahrani. 2022. Impact analysis of keyword extraction using contextual word embedding. In *PeerJ Computer Science*.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. In *biometrics*.

Chantal Lemay, Marie-Claude L'Homme, and Patrick Drouin. 2005. Two methods for extracting "specific" single-word terms from specialized corpora: Experimentation and evaluation. In *International Journal of Corpus Linguistics*.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *EMNLP*.

Harald Lüngen. 2017. Dereko–das deutsche referenzkorpus. In *Zeitschrift für germanistische Linguistik*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*.

Youngja Park, Roy J Byrd, and Branimir Boguraev. 2002. Automatic glossary extraction: Beyond terminology identification. In *COLING*.

Krutarth Patel and Cornelia Caragea. 2021. Exploiting position and contextual word embeddings for keyphrase extraction from scientific papers. In *EACL*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. In *JMLR*.

Yili Qian, Chaochao Jia, and Yimei Liu. 2021. Bert-based text keyword extraction. In *Journal of Physics: Conference Series*.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. In *Text mining: applications and theory*.

Dhruva Sahrawat, Debanjan Mahata, Mayank Kulkarni, Haimin Zhang, Rakesh Gosangi, Amanda Stent, Agniv Sharma, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2019. Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings. In *arXiv*.

Jill Sammet and Ralf Krestel. 2023. Domain-specific keyword extraction using bert. In *Conference on Language, Data and Knowledge*.

Rui Wang, Wei Liu, and Chris McDonald. 2014. Corpus-independent generic keyphrase extraction using word embedding vectors. In *Software engineering research conference*.

Shuting Wang, Chen Liang, Zhaohui Wu, Kyle Williams, Bart Pursel, Benjamin Brautigam, Sherwyn Saul, Hannah Williams, Kyle Bowen, and C Lee Giles. 2015. Concept hierarchy extraction from textbooks. In *ACM Symposium on Document Engineering*.

Christian Wartena. 2019. A probabilistic morphology model for german lemmatization. In *KONVENS*.

Qingsong Wen, Jing Liang, Carles Sierra, Rose Luckin, Richard Tong, Zitao Liu, Peng Cui, and Jiliang Tang. 2024. Ai for education (ai4edu): Advancing personalized education with llm and adaptive learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuan-Jing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on twitter. In *EMNLP*.

Supplementary Material

A Qualitative Examples

This section presents the predictions made by GlossEx (using SUBTLEX-DE as its background corpus) on a specific textbook. The predicted lemmas are divided into two categories: correctly predicted (true positives) and incorrectly predicted (false positives). The results for `Westermann: Kompetenz Politik-Wirtschaft 2006 (Gymnasium Niedersachsen, Stufe 8)` are as follows:

Correct Lemmas	Incorrect Lemmas
Einkommen, Haushalt, Ökonom, Geld, Markt, Wirtschaft, Händler, Anbieter, Knappheit, Angebot, Bedürfnis, Käufer, Nachfrage	Herr, Ergebnis, Person, Wunsch, Cent, Mark, Laden, Mensch, Mitglied, Form, Preis, Verfügung, Mittel, Stand, Kauf, Wochenmarkt, Euro, Prinzip, kaufen, Taschengeld

Our method successfully identifies words with domain-specific meaning, such as "Haushalt" (English: *budget*) and "Nachfrage" (English: *demand*). However, GlossEx also identifies common economic terms that are part of everyday language, like "Laden" (English: *shop*) and "Euro". Additionally, GlossEx finds words such as "Person" (English: *person*) and "Mensch" (English: *human*). These terms have a strong semantic similarity to other human-related words, such as "Käufer" (English: *buyer*), and are therefore included in the list of predicted lemmas. An examination of results from other textbooks generally supports these findings.