information

Article

Automatic Construction of Educational Knowledge Graphs: A Word Embedding-Based Approach

Qurat Ul Ain *, Mohamed Amine Chatti *, Komlan Gluck Charles Bakar, Shoeb Joarder and Rawaa Alatrash

Social Computing Group, Faculty of Engineering, University of Duisburg-Essen, 47057 Duisburg, Germany; komlan.bakar@stud.uni-due.de (K.G.C.B.); shoeb.joarder@uni-due.de (S.J.); rawaa.alatrash@stud.uni-due.de (R.A.)
* Correspondence: qurat.ain@stud.uni-due.de (Q.U.A.); mohamed.chatti@uni-due.de (M.A.C.)

Abstract: Knowledge graphs (KGs) are widely used in the education domain to offer learners a semantic representation of domain concepts from educational content and their relations, termed as educational knowledge graphs (EduKGs). Previous studies on EduKGs have incorporated concept extraction and weighting modules. However, these studies face limitations in terms of accuracy and performance. To address these challenges, this work aims to improve the concept extraction and weighting mechanisms by leveraging state-of-the-art word and sentence embedding techniques. Concretely, we enhance the SIFRank keyphrase extraction method by using SqueezeBERT and we propose a concept-weighting strategy based on SBERT. Furthermore, we conduct extensive experiments on different datasets, demonstrating significant improvements over several state-of-the-art keyphrase extraction and concept-weighting techniques.

Keywords: technology-enhanced learning; massive open online courses; educational knowledge graphs; course knowledge graphs; natural language processing

Citation: Ain, Q.U.; Chatti, M.A.; Bakar, K.G.C.; Joarder, S.; Alatrash, R. Automatic Construction of Educational Knowledge Graphs: A Word Embedding-Based Approach. *Information* **2023**, *14*, 526. https://doi.org/10.3390/info14100526

Academic Editors: Mohamed Hedi Karray, Linda Elmhadhbi, Arkopaul Sarkar and Antonio De Nicola

Received: 16 August 2023
Revised: 15 September 2023
Accepted: 25 September 2023
Published: 27 September 2023

1. Introduction

In recent years, knowledge graphs (KGs), as forms of structured human knowledge, have garnered considerable research attention from both academia and industry [1,2]. A KG, wherein nodes represent entities and edges depict relationships between entities, serves as an integrated information repository that interlinks heterogeneous data from diverse domains [3,4]. KGs have demonstrated robust capability to provide more efficient services in a vast number of application domains, such as recommender systems (Netflix [5]), search engines (Microsoft's Satori and Google's Knowledge Graph [1]), personal assistant apps (Apple's Siri [6]), and question answering (e.g., IBM's Watson [7], Wolfram Alpha [8]), and many more [9]. However, these generic KGs typically do not offer substantial support for many domain-specific applications because they require deep domain information and knowledge [3]. Learning and education is one of such domains. In this research, we focus on the application of KG in the educational domain, referred to as the educational knowledge graph (EduKG). Related to EduKGs are concept maps [3] and concept graphs [10], which are often used in technology-enhanced learning (TEL) systems. These systems, such as massive open online courses (MOOCs) and learning management systems (LMSs), are instrumental in extracting concepts from learning materials and visualizing them to learners in order to help them judge the vital parts of the learning materials [11].

A well-constructed EduKG has many benefits. It can assist learners in easily understanding facts and connecting different concepts, and is pivotal in accurately modeling the knowledge state of learners. Moreover, it can contribute facilitate the provision of more accurate and personalized recommendations of related concepts to be mastered or relevant learning resources to bridge the understanding gap. Previous works on constructing EduKGs typically relied on domain experts to build the KG manually, a process that is time-consuming and costly [12]. Furthermore, the enormous increase in educational data

on TEL platforms necessitates an automatic approach to building EduKGs. Automatic construction of an EduKG is, however, a difficult problem, and brings new challenges, such as accuracy and performance, which require addressing for the effective assembly of EduKGs. To this end, in this paper, we focus on the task of automatically constructing an EduKG in a MOOC platform, taking into consideration the accuracy and performance aspects. The proposed approach can be easily extended to any other TEL platform.

The primary goal of this research is to investigate the potential of using word/sentence embedding techniques to effectively and efficiently construct an EduKG. This research aims to answer the following research question **(RQ):** *How can we leverage knowledge bases and word/sentence embedding techniques to automatically construct an EduKG based on the concepts extracted from learning materials?* To address the accuracy and performance challenges, we propose a pipeline for the automatic, unsupervised construction of EduKGs, relying on state-of-the-art word and sentence embedding techniques. To this end, our methodology effectively combines several methods. These include an unsupervised embedding-based method that extracts keyphrases from the learning materials, a concept identification method that identifies concepts from the keyphrases, a concept expansion method that provides more context to the learners by introducing them to new concepts, a concept-weighting approach that ranks the concepts according to their relevance to the learning materials, and the visualization of the concepts and relationships within an EduKG.

Through this research, we make the following contributions: (1) we adopt and adapt state-of-the-art word/sentence embedding techniques to automatically construct an EduKG, (2) we enhance the SIFRank keyphrase extraction method proposed in [13] by adopting SqueezeBERT [14], a transformer model for word embedding, (3) we propose an embedding-based concept-weighting strategy using the sentence embedding technique SBERT [15], and (4) we conduct empirical studies on different datasets, demonstrating the effectiveness of the SqueezeBERT-enhanced SIFRank keyphrase extraction method as well as the efficiency of SBERT-based concept-weighting strategy against several baselines.

The rest of this paper is organized as follows: Section 2 outlines the theoretical background of this research and discusses related work. Section 3 introduces our MOOC platform, CourseMapper. Section 4 describes our proposed methodology for the automatic construction of EduKGs. Section 5 demonstrates the experimental evaluation and results. Finally, Section 6 concludes the paper and highlights future research directions.

2. Background and Related Work

2.1. Educational Knowledge Graphs

EduKGs are used for various educational purposes, e.g., to support learning and scientific discovery [3], predict prerequisite dependencies among courses in MOOCs [10,16], provide computer-aided education [17], support scientific resource retrieval [18], and recommend learning resources [19], learning paths, knowledge levels [20], and curricula [21]. A variant of EduKG, the course knowledge graph (CKG), integrates scattered courses with knowledge points, and fully reflects the relationship between courses and knowledge points [22,23]. These CKGs are also being widely used to improve learning outcomes, solve problems in traditional teaching, and enhance the possibility of effective learning [23]. Moreover, CKGs are being used as the basis for an intelligent question–answering system for high school courses [24] and course recommendations based on student information [25].

In general, an EduKG is a heterogeneous graph, where nodes denote entities, and edges represent relations between entities. These entities and edges can model different aspects in TEL environments. For instance, these EduKGs usually comprise various entities, namely course concepts or knowledge points [3,26–32], courses [26,29,30,32], course groups [26], course instructors or lecturers [26,32], universities [26,32], websites [26], learning platforms [32], wiki explanations of concepts [27], course sections [28], questions [28], keywords [33], and categories [28,33]. In general, it can be concluded that concepts or knowledge points are the basic building blocks of every EduKG and, hence, must be extracted accurately and efficiently.

Previously, EduKGs were manually constructed by course experts. However, due to the limitations of this manual approach in terms of time and effort, different works in recent years have addressed the problem of automatically generating EduKGs by employing machine learning (ML) and natural language processing (NLP) techniques. Various methods have been introduced by researchers depending on the requirements and tasks to be achieved. The most essential task in the construction of EduKGs is **entity extraction**. This task has been achieved based on a variety of ML and NLP-based methods and will be discussed in detail in Section 2.2. Another important task in the construction of EduKGs is **relation extraction,** which involves connecting different entities in the EduKG. These relationships are mainly sequential or semantic relationships and are found using DBpedia spotlight [25], rule processing, KNN [26], probabilistic association rule mining [3], binary classification on models trained with labeled datasets [34], binary classification on trained models using feedforward neural networks [35], prompt-tuning with synergistic optimization [36], cosine similarity [27,29], PMI, normalized google distance [31], semantic role labeling [37], prerequisite relation calculations based on the preliminary knowledge tags of the courses [20], and rule-based relation extraction [38]. Another widely adopted task in the construction of EduKGs is **entity linking** with an external knowledge base, mainly Wikipedia [25,27,28,32,33,39]. This is achieved using Wikipedia API or different entity-linking services, such as DBpedia Spotlight [25,39] and Babelfy [39]. Some of the research studies have also focused on **concept expansion** for the semantic enrichment with new related concepts, for example, using Wikipedia categories [25,33,39] or the Baidu Encyclopedia [31]. Furthermore, some of the studies have also incorporated **concept-weighting** strategies to rank the identified concepts based on their weights. The weighting strategy used in most of the studies is term frequency–inverse document frequency (TF-IDF) [25,26,37,39].

Most related works primarily focus on a subset of the tasks mentioned above. Only [25] adopted a more complete set of steps, including concept linking, expansion, and weighting. Unlike the EduKG construction pipeline in [25], we perform keyphrase extraction before the concept linking step. This step is introduced to the pipeline to achieve better efficiency and performance by annotating only the extracted keyphrases with the entity-linking service rather than sending the whole text of the learning material for annotation. Furthermore, in the concept-weighting step, we use word embedding techniques instead of TF-IDF. Although efficient, TF-IDF-based concept-weighting methods are usually imprecise because they do not consider the semantic relatedness between concepts.

2.2. Automatic Concept Extraction

Concepts represent the core part of any EduKG. Thus, **concept extraction** is an essential step in the construction of EduKGs. To obtain concept-level knowledge, manual indexing of learning materials or textbooks is challenging, time-consuming, and prone to errors [40]. There has been increased research on automatic concept extraction from learning materials, mainly through direct entity linking of the materials to external knowledge bases, or by performing keyphrase extraction on the text of the learning materials.

Several research studies have used direct entity linking via named entity recognition (NER) [38,41,42] and entity-linking services [25,26,28,32,33,39,41,43]. It is a common practice to use entity linking to identify concepts from the text. However, applying entity linking to learning materials with a large amount of text is not efficient, as this step requires sending multiple requests to an entity-linking service. To overcome this issue, few works have focused on first extracting keyphrases from the text and then using these keyphrases to identify concepts. These works have employed various machine learning methods, such as rule-based learning [44], supervised learning [40], unsupervised learning [37], and deep neural networks [3,31].

In our work, we apply **keyphrase extraction** as a pre-step to entity linking from Wikipedia. Our aim is to filter the list of concept candidates, thus making the automatic concept extraction more efficient. We focus on unsupervised methods for keyphrase

extraction, as the supervised methods need models trained on the domain-specific corpus to detect concepts, requiring a significant amount of training data, which are not available in our context. We adopt, enhance, and compare a state-of-the-art word embedding-based keyphrase extraction approach with several baselines. To the best of our knowledge, it is the first work applying word embedding-based keyphrase extraction methods for educational concept extraction. Other works used word embeddings to construct an EduKG but for a purpose other than automatic concept extraction [20,27].

3. EduKG in CourseMapper

Our main goal is to automatically construct an EduKG for the learning material in CourseMapper. CourseMapper is a MOOC platform that we developed to help learners effectively create and manage courses, collaborate with each other, and interact with learning materials using shared annotations [45]. Generally, an EduKG on top of a MOOC platform is a heterogeneous information network consisting of different entities and relationships between them. Concepts represent important entities to be modeled in any EduKG in order to provide an overview of the main concepts in the learning materials, help learners be aware of what is to be learned, and draw their attention to the most crucial concepts. Consequently, this facilitates the learners' achievement and creates a positive perception of the materials [46]. In addition to concepts, the EduKG normally contains other entities, such as learners, teachers, courses, and videos, as well as the relationships among those entities. Figure 1 shows a part of the network schema of our EduKG in CourseMapper with different entity types and the semantic links between them. This part of the EduKG consists of five types of entities, namely the user, learning material (slides or video), concept, related concept, and category, as well as the relationships between these entities.

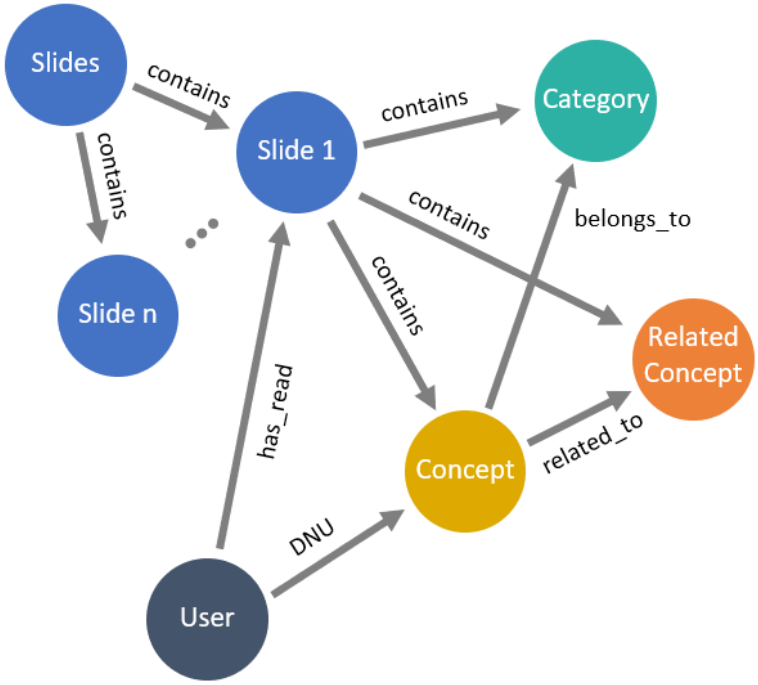

Figure 1. Network schema representing the EduKG entities and the relationships between them.

With every learning material (slides in our example), learners are provided with a "Did Not Understand (DNU)" button, which they can click to view the concepts extracted from the current slide (see Figure 2). Then, they can click to select the concrete concepts that they do not understand from that slide. This information is used to model the relationships "Understand" (U) and "Did-Not-Understand" (DNU) between learners and concepts in the EduKG. This can provide an effective way to model the learner's knowledge state,

which can be used to provide personalized recommendations of prerequisite concepts to be mastered, as well as relevant related external learning resources that can help learners understand the concepts on the slide. The concepts extracted from the learning materials are expanded based on related concepts and categories in Wikipedia (see Figure 1). In the next section, we will discuss our proposed approach for the automatic extraction of concepts in the EduKG.

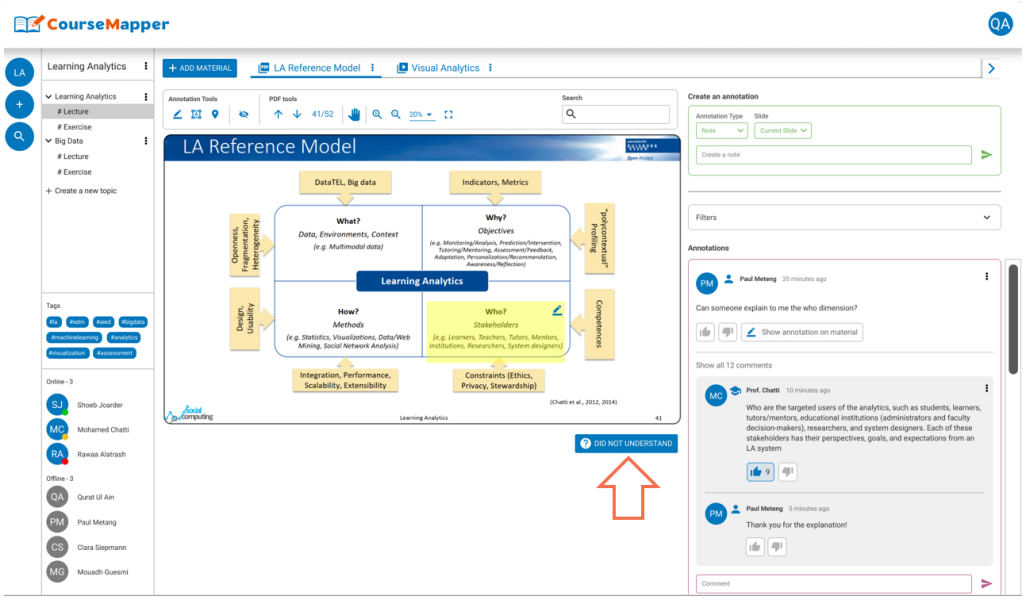

Figure 2. CourseMapper's user interface, showing the DNU button.

4. Methodology

Regarding the development of the KG for learning materials, the following essential factors must be considered: The KG aims to highlight the relevant concepts inside the learning materials, as well as allow learners to discover new concepts that are not necessarily mentioned in the learning materials. This section describes the pipeline we used for the automatic extraction of concepts to build the EduKG in CourseMapper (see Figure 3); we specifically focus on the keyphrase extraction and concept-weighting modules, which are the core contributions of this paper. Note that, although we are building an EduKG in a MOOC platform, all the components of our pipeline can be replicated for any other TEL environment.

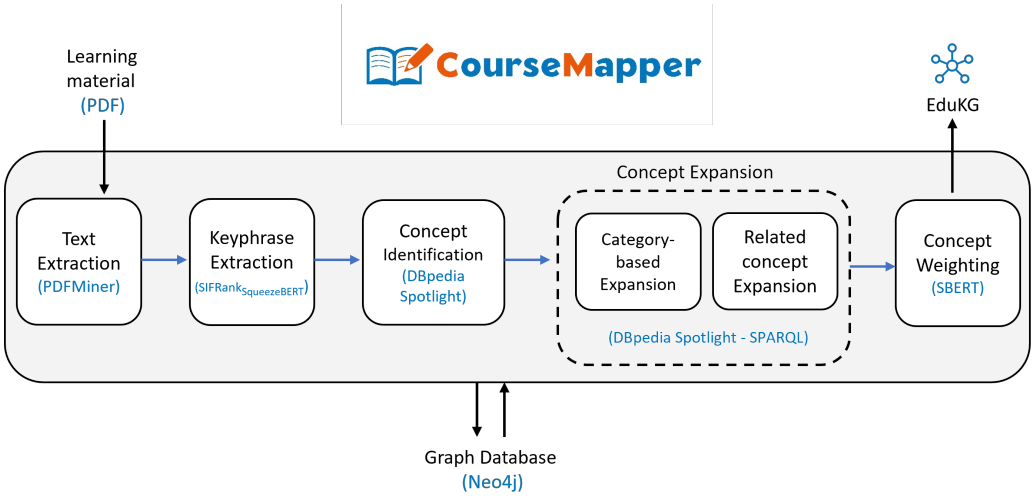

Figure 3. Pipeline for the automatic construction of an EduKG.

4.1. Text Extraction

The first step in the construction of our EduKG is text extraction from PDF learning materials in CourseMapper. Extracting text effectively and accurately from the PDF is one of the most challenging tasks in text mining because of the variation in text styles and formats. Usually, the extraction method lacks awareness of the layout of the PDF file. To solve this problem, the layout aspect of the PDF file has to be taken into consideration to make sure that words in the same box are accurately extracted. We used a simple form of the method proposed in [47], a layout-aware PDF text extraction. The method consists of three steps: (1) detecting contiguous text blocks, (2) classifying text blocks into categories using a rule-based method, and (3) stitching classified text blocks together in the correct order. For the first step, contiguous text blocks are detected from the PDF using an open-source tool PDFMiner [48]. With PDFMiner, information about characters and their positions in the PDF file is obtained. Then as in step (3) of the mentioned approach [47], based on the proximity of their coordinates, characters are grouped into lines and blocks of text.

4.2. Keyphrase Extraction

We applied keyphrase extraction as a pre-step to entity linking from Wikipedia. This step is motivated by the need to avoid sending learning materials with large amounts of text to an entity-linking service, thus it improves the efficiency of the next step in the pipeline, i.e., concept identification (see Section 4.3). The keyphrase extraction step is built on the basis of SIFRank [13], a state-of-the-art unsupervised keyphrase extraction method that is based on a pre-trained language model. The first step in the SIFRank process is text preprocessing. Tokenization is performed on the text using the CoreNLP library. Then, part-of-speech (POS) tagging is applied to the generated tokens using the same library. Lastly, embeddings of these POS-tagged tokens are generated using embedding techniques. The authors in [13] created word embeddings of the tokens with ELMo [49], and sentence embeddings with sentence inverse frequency (SIF) [50]. According to the authors in [13], SIFRank achieved the best performance in keyphrase extraction for short documents. For long documents, the authors extended SIFRank to SIFRankplus, which uses position-biased weight to improve its performance in long documents.

We adopted the SIFRank keyphrase extraction process with some modifications to achieve better results, based on our experiments. Firstly, we performed tokenization on the text. Then, we applied POS tagging on the generated tokens. Lastly, we generated embeddings of these POS-tagged tokens using the SIFRank and SIFRankplus embedding techniques for short and long documents, respectively. Recognizing the performance of state-of-the-art transformer-based pre-trained language models, we replaced ELMo with the pre-trained SqueezeBERT [14] as a word embedding method in SIFRank and SIFRankplus. The decision to use SqueezeBERT is motivated by its lightweight transformer architecture with higher information flow between the layers; moreover, it is faster than the BERT model [14]. We conducted extensive experiments using the SIFRank/SIFRankplus with ELMo-based and SIFRank/SIFRankplus with SqueezeBERT-based pre-trained models and compared them with different statistical, graph-based, and embedding-based keyphrase extraction methods in terms of accuracy (see Section 5.1). The results show that while our proposed SIFRank/SIFRankplus with SqueezeBERT methods outperformed other baseline models, the accuracy of the keyphrase extraction task was relatively low with an F1-score of 40.38% in the best case.

4.3. Concept Identification

The keyphrases obtained in the previous step are used as input for the concept identification step. Identifying concepts from the keyphrases requires the use of external knowledge bases. Similar to [25], we use DBpedia Spotlight [51] to link keyphrases to concepts in DBpedia. DBpedia Spotlight is a powerful entity-linking tool that can quickly and efficiently identify meaningful substrings (or annotations) in a text and link them to related DBpedia concepts. DBpedia Spotlight is used for automatically annotating

mentions of DBpedia resources in text, based on three steps: (1) spotting, i.e., applying a string-matching algorithm based on the DBpedia lexicon to identify any mention in the text matching a DBpedia resource, with priority given to the longest case-insensitive match; (2) candidate selection, i.e., selecting candidate DBpedia resources for each spotted mention to narrow down the space of disambiguation possibilities; and (3) disambiguation, i.e., finding the best candidate DBpedia resource for the spotted mention using contextual information. We use the DBpedia Spotlight web service, which supplies endpoints for spotting, disambiguation, and annotation by keeping the support parameter set to 5 and the confidence parameter to 0.35, as used by [39]. We also use the contextual score assigned by DBpedia Spotlight to a candidate DBpedia resource using its contextual information [51].

Concepts can, however, wrongly be identified when using DBpedia Spotlight or any other entity-linking services. Manrique et al. [25] refer to this problem as incorrect annotation, i.e., "the concepts are mistakenly linked". Since this is an automatic process and, thus, the identified concepts are not manually evaluated, we use a weighting strategy that looks at the semantic similarities of the identified concepts to the learning materials. This weighting strategy is discussed in detail in Section 4.5.

4.4. Concept Expansion

To enrich the EduKG, we apply an expansion step to the identified concepts. Considering that the keyphrase extraction step does not produce results with high accuracy (see Section 4.2) and, consequently, the concept identification step does not determine all concepts relevant to the learning materials, we expand the EduKG with additional concepts. These additional concepts can improve the coverage and diversity of the identified concepts, enhance the structure of the EduKG, help further develop the learners' knowledge, and promote concept exploration and discovery. Expanded concepts can also be important to reinforce the main topic of the learning material if they are not present in the learning material [52]. The expansion is performed while considering the semantic relationships of the concepts in the knowledge base. We adopt the two types of expansion, namely category-based expansion and property-based expansion, as proposed in [25]. We refer to property-based expansion as *a related concept expansion* in this work.

4.4.1. Category-Based Expansion

The EduKG is enriched with the categories of the identified concepts. For example, for the concept of "Natural language processing", the categories "Category:Computational linguistics" and "Category:Artificial intelligence" are added to the EduKG. These added categories provide hierarchical information about the concepts and allow users to discover more broad concepts. SPARQL queries retrieving the categories (through the ontology property dct:subject) of the concepts are executed against the public SPARQL endpoint of DBpedia.

4.4.2. Related Concept Expansion

The EduKG is enriched with the related concepts of the identified concepts. For example, for the concept of "Natural language processing", the related concept of "Natural language understanding" is added to the graph. The related concepts are extracted using SPARQL and the ontology property dbo:wikiPageWikiLink. To determine the existence of a path between the two concepts, SPARQL queries are executed against the public SPARQL endpoint of DBpedia to retrieve the property paths between the two concepts. An edge is created between the two nodes if a property path is found.

4.5. Concept Weighting

The expansion is a beneficial step in creating the EduKG; however, the expansion can introduce even more noisy concepts in the graph [53]. To overcome this problem, the concept-weighting strategy is used to weight the concepts in the graph. With this weighting

strategy, concepts that are contextually and semantically similar to the learning materials will have high weights as opposed to noisy concepts whose weights will be lower.

Manrique et al. [25] proposed a weighting strategy to solve the aforementioned problem, based on three steps: (1) concept frequency: a TF-IDF-based weighting of the concepts in the KG, which helps determine the importance of a concept to a particular piece of learning material by penalizing concepts that appear often across multiple learning materials; (2) category discount: this penalizes categories that are too broad and generic in the hierarchy of the KG; and (3) related concept discount: this penalizes frequently related concepts. This strategy, albeit effective as a weighting strategy, fails to consider the context of the words and capture the semantic awareness from both the learning materials and the concepts. Recently, word and sentence embedding techniques have garnered increasing attention due to the impressive performance they have demonstrated across a wide range of NLP-related scenarios. These techniques excel at capturing the semantic meaning of words or documents and the contextual relationships between them, which can be effectively used to extract meaningful data representations, gain a semantic and relational understanding of the data, and measure semantic similarities between words or documents [54,55]. For this reason, we propose another weighting strategy for the concepts by leveraging the advantages of transformer models, particularly the SBERT model [15]. Our approach (w_{SBERT}) works as follows.

4.5.1. Concept Weighting

The embedding of the content of the learning material emb_{lm_i} and the embedding of the content of the concept (i.e., the text content of its Wikipedia article) emb_c are defined using SBERT. To retrieve the Wikipedia article text, the Wikipedia API (https://pypi.org/project/Wikipedia-API/, accessed on 1 August 2023) is used. Then, the weight of the concept in the learning material is computed as the cosine similarity score between emb_{lm_i} and emb_c.

4.5.2. Related Concept Weighting

The embedding of the content of the learning material emb_{lm_i} and the embedding of the content of the related concept (the text content of the Wikipedia article) emb_{rel} are computed using SBERT. The weight of the related concept is the cosine similarity score between emb_{lm_i} and emb_{rel}.

4.5.3. Category Weighting

The embedding of the content of the learning material emb_{lm_i} and the embedding of the category name emb_{cat} are computed using SBERT. Wikipedia category pages are not Wikipedia articles and, thus, do not have text content. Therefore, the name of the category is used. The weight of the category is the cosine similarity score between emb_{lm_i} and emb_{cat}. This will help determine the category's importance in the learning material.

4.6. Knowledge Graph Storage

A graph database is highly suitable for storing the KG data since the concepts will need to be stored with several properties. The constructed KG is stored in a Neo4j graph database as follows:

- Each concept in the KG is stored as a node. The concept is attached to properties, such as the name of the concept.
- A relationship of the type *related_to* is stored to connect two nodes, either a concept from the identification module or a related concept from the expansion module.
- A relationship of the type *belongs_to* is stored to connect a node to a node of the type category.
- The learning material is stored as a node.
- A relationship of the type *contains* is stored to connect the learning material to the concepts.

Figure 4 shows the illustration of the KG stored in a graph database. There are four types of nodes to be observed: The learning material *LM1*, which contains a concept *C1*, a related concept *C2* and a category *C3*. Supposing that following the expansion and weighting steps, concept *C1* is found to be related to *C2* and is categorized under *C3* and, thus, added to the KG, relationships of the type *related_to* and *belongs_to* are created between *C1* and *C2*, and between *C1* and *C3*, respectively.

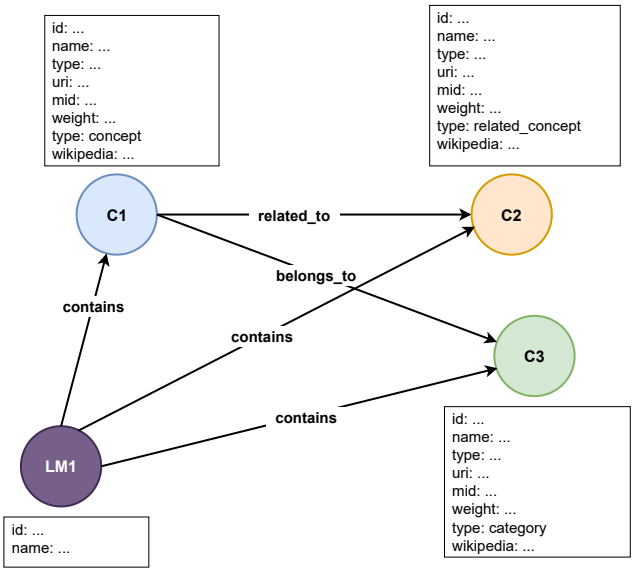

Figure 4. KG illustration in a graph database.

4.7. Knowledge Graph Visualization

The KG is visualized in CourseMapper by showing a button at the top of each learning material, to make it easily accessible to users, as shown in Figure 5. With this button, users are able to display or hide the KG interface. As illustrated in Figure 6, users are also able to interact with the KG through selection and filtering actions, to only view specific parts of the KG, based on their needs. Additionally, users can obtain more information on a concept by selecting its node. Once the node is selected, the Wikipedia abstract of the concept will appear on the right side of the KG interface. If needed, users can also see a detailed description of the concept by clicking the "Read full article on Wikipedia" button (see Figure 6).

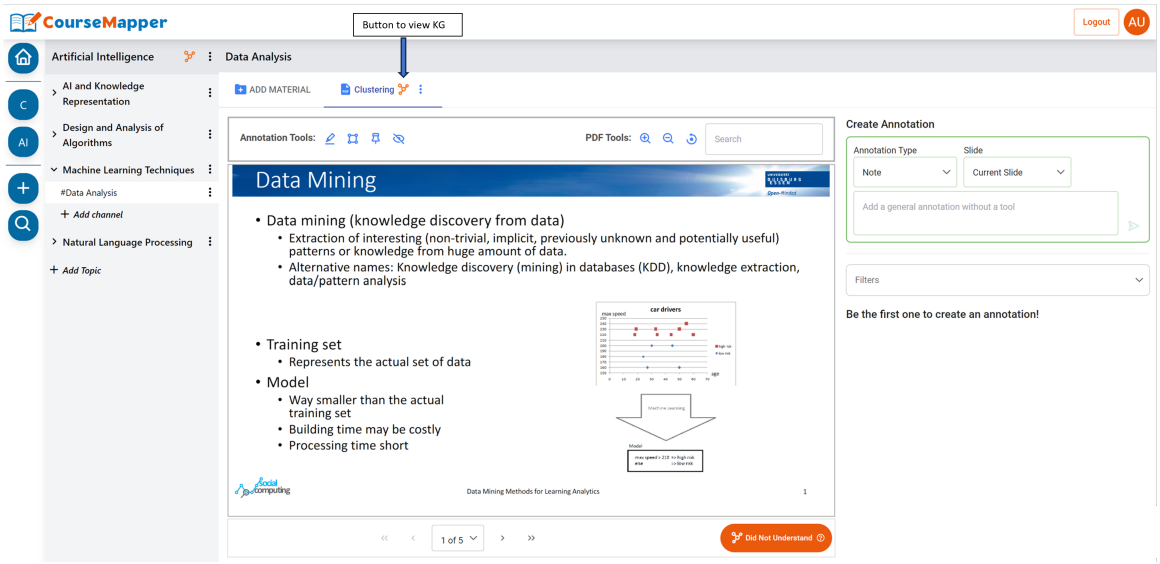

Figure 5. Learning material with the KG button.

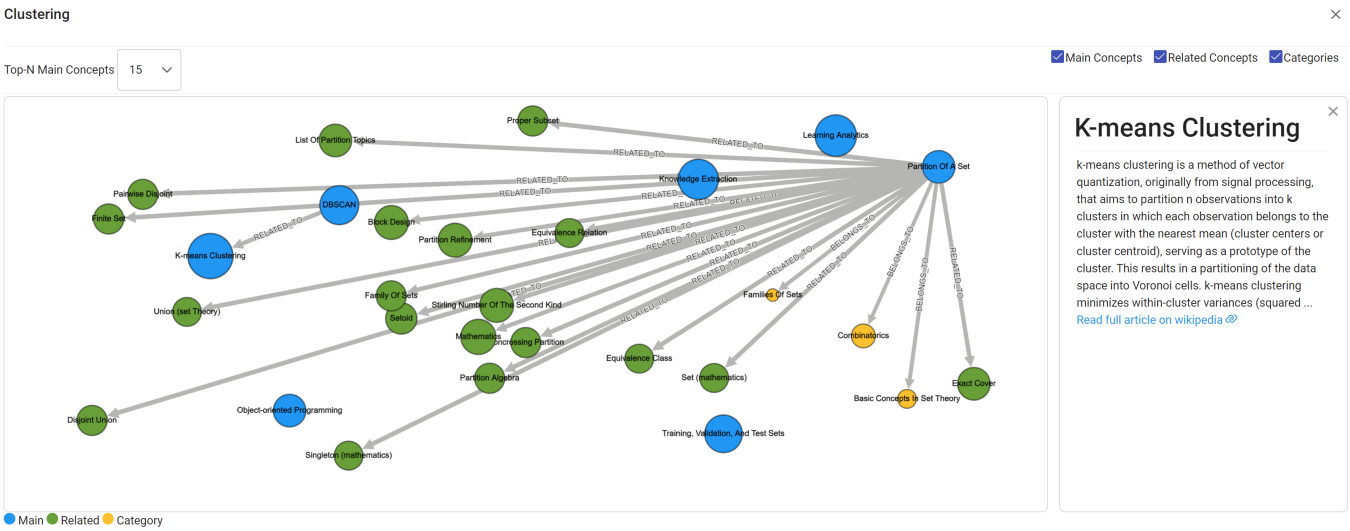

Figure 6. Visualizing and interacting with a KG in CourseMapper.

5. Experimental Evaluation

Experiments are carried out to evaluate the performance of the proposed methods related to the keyphrase extraction and concept-weighting steps of the EduKG construction pipeline.

5.1. Keyphrase Extraction Evaluation

We compare the extraction accuracies and efficiencies of different keyphrase extraction methods, such as statistical, graph-based, and embedding-based keyphrase extraction methods. The embedding-based methods include the original SIFRank and SIFRankplus methods based on ELMo [13], and our proposed keyphrase extraction methods, SIFRank and SIFRankplus, based on SqueezeBERT for short and long documents, respectively.

5.1.1. Datasets

In order to conduct the experiments, three well-known benchmark datasets are used: (1) Inspec [56], a document collection of 2000 scientific abstracts with sets of keyphrases identified by expert annotators, (2) SemEval2017 [57], a double annotated document collection of 493 paragraphs extracted from 500 ScienceDirect journal articles, and (3) DUC2001 [58], a collection of 308 news articles collected from TREC-9. Table 1 presents a summary of the datasets.

Table 1. Summary of datasets.

Dataset	Inspec	SemEval2017	DUC2001
Type of Documents	Abstracts	Paragraph	News
No. of Documents	500	493	308
Average Words	134.4	194.7	828.4
Average keyphrases	9.8	17.3	8.1

5.1.2. Baselines

The proposed keyphrase extraction methods ($SIFRank_{SqueezeBERT}$ and $SIFRankplus_{SqueezeBERT}$) are compared against different unsupervised keyphrase extraction methods on the selected datasets. These methods include statistical-based methods (YAKE [59], RAKE [60], and TF-IDF [61]), graph-based methods (MultiPartiteRank [62], TopicalPageRank [63], TopicRank [64], PositionRank [65], SingleRank [66], TextRank [67]), embedding-based models

(EmbedRank [68], SIFRank, and SIFRankplus [13]). EmbedRank is evaluated using sent2vec ($s2v$) and doc2vec ($d2v$). The baselines generate candidate phrases using noun phrases. The Python keyphrase extraction (PKE) (https://github.com/boudinfl/pke, accessed on 1 August 2023) is used to run the statistical-based models and the graph-based models. EmbedRank (https://github.com/swisscom/ai-research-keyphrase-extraction, accessed on 1 August 2023) and SIFRank (https://github.com/sunyilgdx/SIFRank, accessed on 1 August 2023) are used to produce the results of the embedding-based models on the selected datasets.

5.1.3. Results and Analysis

We conducted experiments using the above-mentioned baseline models and the proposed $SIFRank_{SqueezeBERT}$ and $SIFRankplus_{SqueezeBERT}$ methods on the introduced short document datasets (Inspec and SemEval2017) and long document dataset (DUC2001), to extract the top 5, 10, and 15 keyphrases. The metrics used for evaluation were precision, recall, and F1-score. The results are presented in Table 2. Here, metrics are denoted as percentage points and bold indicates the best metric.

Table 2. Evaluation results of keyphrase extraction methods. K is the number of keyphrases extracted from a single document by the models.

K	Method	Inspec			SemEval2017			DUC2001		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
5	*YAKE*	22.48	11.44	15.16	25.15	7.27	11.28	-	-	-
	RAKE	30.33	15.43	20.45	29.61	8.56	13.28	-	-	-
	TFIDF	16.52	8.41	11.14	30.06	8.69	13.48	-	-	-
	MultiPartiteRank	28.94	14.72	19.51	34.40	9.94	15.43	-	-	-
	TopicalPageRank	32.96	16.77	22.23	35.74	10.33	16.03	-	-	-
	TopicRank	28.66	14.57	19.32	33.95	9.81	15.23	-	-	-
	PositionRank	31.98	16.27	21.56	35.82	10.35	16.06	-	-	-
	SingleRank	31.52	16.04	21.26	34.04	9.84	15.26	-	-	-
	TextRank	17.85	8.81	11.80	18.33	5.30	8.22	-	-	-
	$EmbedRank_{d2v}$	26.36	13.41	17.78	29.24	8.45	13.11	9.70	6.00	7.42
	$EmbedRank_{s2v}$	40.24	20.48	27.14	47.74	13.79	21.41	33.81	20.91	25.84
	$SIFRank_{ELMo}$	43.20	21.99	29.14	48.64	14.06	21.81	31.79	19.67	24.30
	$SIFRankplus_{ELMo}$	42.12	21.44	28.41	47.99	13.87	21.52	40.26	24.91	30.78
	$SIFRank_{SqueezeBERT}$	**44.00**	**22.39**	**29.68**	**49.21**	**14.22**	**22.07**	30.75	19.02	23.50
	$SIFRankplus_{SqueezeBERT}$	-	-	-	-	-	-	**41.69**	**25.80**	**31.87**
	Improvement (%)	1.85	1.81	1.85	1.17	1.14	1.19	3.51	3.57	3.54
10	*YAKE*	18.00	18.32	18.16	22.74	13.14	16.66	-	-	-
	RAKE	27.60	27.89	27.74	28.42	16.43	20.82	-	-	-
	TFIDF	14.62	14.88	14.75	24.16	13.96	17.70	-	-	-
	MultiPartiteRank	24.19	24.28	24.24	29.00	16.75	21.24	-	-	-
	TopicalPageRank	29.60	29.86	29.73	33.04	19.10	24.21	-	-	-
	TopicRank	24.09	23.92	24.00	27.41	15.84	20.08	-	-	-
	PositionRank	27.45	27.62	27.54	31.87	18.42	23.34	-	-	-
	SingleRank	28.77	29.11	28.94	32.45	18.76	23.77	-	-	-
	TextRank	14.81	12.70	13.68	17.31	9.72	12.45	-	-	-
	$EmbedRank_{d2v}$	24.90	25.02	24.96	27.44	15.86	20.10	8.91	11.00	9.85
	$EmbedRank_{s2v}$	34.96	35.09	35.03	42.88	24.78	31.41	28.05	34.62	30.99
	$SIFRank_{ELMo}$	38.63	38.84	38.73	43.45	25.11	31.83	24.97	30.83	27.60
	$SIFRankplus_{ELMo}$	36.44	36.64	36.54	42.49	24.56	31.13	30.20	37.28	33.37
	$SIFRank_{SqueezeBERT}$	**39.36**	**39.58**	**39.47**	**43.73**	**25.28**	**32.04**	26.04	32.16	28.78
	$SIFRankplus_{SqueezeBERT}$	-	-	-	-	-	-	**31.79**	**39.26**	**35.13**

Table 2. *Cont.*

K	Method	Inspec			SemEval2017			DUC2001		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
	Improvement (%)	1.89	1.9	1.91	0.64	0.67	0.65	5.26	5.31	5.27
15	*YAKE*	16.10	24.59	19.46	21.73	18.84	20.18	-	-	-
	RAKE	24.34	36.20	29.11	25.79	22.35	23.94	-	-	-
	TFIDF	13.88	21.19	16.77	20.76	18.00	19.28	-	-	-
	MultiPartiteRank	22.16	32.14	26.23	25.95	22.45	24.07	-	-	-
	TopicalPageRank	26.65	39.15	31.71	30.52	26.44	28.33	-	-	-
	TopicRank	21.49	30.09	25.08	24.02	20.68	22.23	-	-	-
	PositionRank	24.31	35.50	28.86	29.12	25.21	27.02	-	-	-
	SingleRank	25.71	38.05	30.69	29.84	25.85	27.70	-	-	-
	TextRank	13.00	13.64	13.31	15.80	12.05	13.68	-	-	-
	$EmbedRank_{d2v}$	23.58	34.30	27.95	26.90	23.28	24.96	8.31	15.31	10.77
	$EmbedRank_{s2v}$	31.70	45.98	37.53	38.43	33.26	35.66	24.12	44.45	31.27
	$SIFRank_{ELMo}$	33.49	48.76	39.71	39.14	33.88	36.32	21.56	39.74	27.95
	$SIFRankplus_{ELMo}$	32.64	47.52	38.70	38.61	33.43	35.83	24.86	45.83	32.24
	$SIFRank_{SqueezeBERT}$	**34.06**	**49.59**	**40.38**	**39.78**	**34.43**	**36.91**	23.06	42.52	29.91
	$SIFRankplus_{SqueezeBERT}$	-	-	-	-	-	-	**26.91**	**49.62**	**34.90**
	Improvement (%)	1.7	1.7	1.68	1.63	1.62	1.63	8.24	8.26	8.25

These methods include statistical-based methods (YAKE [13], RAKE [44], and TF-IDF [46]), graph-based methods (MultiPartiteRank [11], TopicalPageRank [28], TopicRank [12], PositionRank [22], SingleRank [52], and TextRank [35]), and embedding-based models (EmbedRank [8], SIFRank, and SIFRankplus.

The results show that embedding-based models (EmbedRank, SIFRank, SIFRankplus) perform better than statistical-based (YAKE, RAKE, TF-IDF) and graph-based models (MultiPartiteRank, TopicalPageRank, TopicRank, PositionRank, SingleRank, TextRank) in all datasets.

Our proposed $SIFRank_{SqueezeBERT}$ and $SIFRankplus_{SqueezeBERT}$ methods (marked in bold in Table 2) consistently yield the best performances compared with other baseline models in short document datasets (Inspec and SemEval2017) and the long document dataset (DUC2001), respectively. In particular, $SIFRank_Plus_{SqueezeBERT}$ demonstrates improvement over the strongest baseline $SIFRankplus_{ELMo}$ (underlined in Table 2) w.r.t precision by 3.51%, recall by 3.57%, and F1-score by 3.54% when extracting 5 keyphrases. When extracting 10 keyphrases, the improvements are 5.26% w.r.t precision, 5.31% w.r.t recall, and 5.27% w.r.t the F1-score. Lastly, $SIFRank_Plus_{SqueezeBERT}$ demonstrates improvement over the strongest baseline $SIFRankplus_{ELMo}$ w.r.t precision by 8.24%, recall by 8.26%, and F1-score by 8.25% when extracting 15 keyphrases. This indicates that using SIFRank combined with SqueezeBERT allows for better and more accurate extraction of keyphrases from documents in comparison to SIFRank based on ELMo.

Referring to the execution time, SIFRank with SqueezeBERT is 6x faster than SIFRank with ELMo at extracting the keyphrases from the Inspec dataset, 4x faster at extracting keyphrases from the SemEval2017 dataset, and almost 3x faster at extracting from the DUC2001 dataset. This shows that ELMo, which is an LSTM language model, is slower than SqueezeBERT.

Furthermore, we conducted experiments to evaluate the performances of different layers of SqueezeBERT. SqueezeBERT originally had 12 layers, and the experiments aimed to find the layers that performed best on the keyphrase extraction task. The experiment was performed by activating and deactivating certain layers of SqueezeBERT each time before computing the word embeddings of the texts. The results of the experiments are shown in Table 3. To save space, we only present a few results in the table. The layers are referred to as ALL (all layers are utilized for word embeddings, and the resultant embeddings are derived from the averaged output of all 12 layers), LX (e.g., L0 represents the first

layer) and LX_Y, i.e., the word embeddings are the average of the output of the X and Y layers. The results indicate that out of all 12 layers, layers 3 and 5 combined delivered the best performance results (marked bold in the table) on short documents (Inspec and SemEval2017 datasets). When layers 3 and 5 were used individually, the performance deteriorated as they were not capable of efficiently capturing the semantic context of the words. In long documents (DUC2001 dataset), layers 0 and 1 outperformed other layers (marked bold in the table). These layers combined can efficiently capture the contextual features of words in long documents.

Table 3. Performance evaluation of SqueezeBERT layers on different datasets.

K	Method	Layers	Inspec			SemEval2017			DUC2001		
			P	R	F1	P	R	F1	P	R	F1
5	*SIFRank−SqueezeBERT*	ALL	43.16	21.97	29.11	48.23	13.94	21.63	-	-	-
		L3	43.0	21.88	29.00	49.13	14.20	22.03	-	-	-
		L3_5	**44.0**	**22.39**	**29.68**	**49.21**	**14.22**	**22.06**	-	-	-
5	*SIFRankplus−SqueezeBERT*	L0	-	-	-	-	-	-	41.17	25.47	31.47
		L1	-	-	-	-	-	-	39.93	24.70	30.52
		L0_1	-	-	-	-	-	-	**41.69**	**25.79**	**31.87**

5.2. Concept-Weighting Evaluation

The experimental evaluation of the concept-weighting step is presented to show the performance of our proposed SBERT-based concept-weighting method, referred to as w_{SBERT} (see Section 4.5), as compared to the baselines.

5.2.1. Experimental Setup

To evaluate the proposed concept-weighting method against the baselines, we use the DBpedia 2016-10 release version, which is one of the most used KG for NLP research. We further use DBpedia Spotlight in the concept identification step to annotate the keyphrases. The EduKG constructed in this research is to be applied in CourseMapper, which is a MOOC platform. Therefore, the CCI dataset (https://github.com/Ruframapi/CCI, accessed on 1 August 2023) is used as the evaluation dataset. It consists of 96 video transcripts of learning resources extracted from Coursera (https://www.coursera.org/, accessed on 1 August 2023) in the area of programming fundamentals. For all learning resources, the dataset contains the core concepts, which were annotated by seven experts [25]. In the keyphrase extraction step, the top 15 keyphrases are extracted from the learning resources. The same evaluation metrics used to evaluate the keyphrase extractions, i.e., precision, recall, and F1-score, are used to determine the accuracies of the top k ranked concepts.

5.2.2. Baselines

Several weighting strategies are used as baselines to compare the accuracy of our proposed w_{SBERT} concept-weighting method. The first strategy, referred to as w_{cf}, uses simple term frequency (TF) to weight the concepts [69]. The second strategy, referred to as w_{cf-idf}, involves the enhancement of the TF-based strategy by adding inverse document frequency (IDF) with a discount for expanded concepts, as proposed by Manrique et al. [25]. The other strategies use centrality measures [70], such as (1) degree centrality (DE): the weight of a node (concept or related concept or category), c, is the number of nodes connected to c divided by the total number of nodes in the KGl; (2) betweenness centrality (BET): the weight of a node, c, is the fraction of the shortest paths between all possible node pairs that pass through c; (3) PageRank (PR): it ranks the importance of nodes in the KG; PR estimates that a node ranks high if the sum of the ranks of its backlinks is high. The weight of a node, c, is the PR score.

5.2.3. Results and Analysis

Figure 7 shows the distribution of concepts in the CCI dataset when we map the experts' annotated concepts to the concepts, related concepts, and categories identified by our approach. Most of the concepts annotated by the experts are related concepts (183). None of the annotated concepts in the dataset is of the category type; 98 concepts (unknown concepts) annotated by the experts have not been identified after our concept identification and expansion steps. Based on the results obtained during the experiments for the top 3, 5, and 10 ranked concepts, it is observed that our strategy w_{SBERT} (marked in bold in Table 4) is more precise than the other strategies, as the precision for all top k-ranked concepts is higher (see Table 4, without harmonic mean). In particular, for the top 10 ranked concepts, w_{SBERT} demonstrates improvement over the strongest baselines (underlined in Table 4) w.r.t precision by 17.8% and w.r.t the F1-score by 14.5%. This shows that w_{SBERT} is the better strategy for selecting a high number of concepts. Despite the higher precision of our strategy, w_{SBERT}, PR shows better recall and an F1-score for the top three and top five ranked concepts. This can be explained by the fact that during the annotation of concepts, the experts focused more on the diversity of the concepts in the learning resource collection. Another explanation for these results is that, while in our approach, concepts were identified from only the top 15 keyphrases extracted from the learning resource, Manrique et al. [25] did not use an intermediate keyphrase extraction module in their experiments but directly fed the learning resource content to the DBpedia Spotlight, which resulted in a higher count of identified concepts.

Table 4. Evaluation results of concept-weighting techniques for the top k-ranked concepts.

K	Method	Without Harmonic Mean			With Harmonic Mean		
		P	**R**	**F1**	**P**	**R**	**F1**
3	w_{cf}	23.6	20.5	21.9	23.6	20.5	21.9
	w_{cf-idf}	<u>24.2</u>	25.4	24.8	<u>24.2</u>	25.4	24.8
	DE	18.5	17.1	17.8	18.5	17.1	17.8
	BET	19.4	20.3	19.8	19.4	20.3	19.8
	PR	23.9	<u>30.1</u>	<u>26.6</u>	23.9	<u>30.1</u>	<u>26.6</u>
	$w_{SBERT}(Ours)$	**24.3**	20.7	22.3	23.9	20.3	22.0
	Improvement (%)	0.4	-	-	-	-	-
5	w_{cf}	17.9	25.7	21.1	17.9	25.7	21.1
	w_{cf-idf}	<u>19.2</u>	27.2	22.5	<u>19.2</u>	27.2	22.5
	DE	14.9	20.1	17.1	14.9	20.1	17.1
	BET	16.7	26.3	20.4	16.7	26.3	20.4
	PR	18.7	<u>35.1</u>	<u>24.4</u>	18.7	<u>35.1</u>	<u>24.4</u>
	$w_{SBERT}(Ours)$	**20.0**	28.3	23.4	**21.3**	30.2	**25.0**
	Improvement (%)	4.1	-	-	11	-	2.5
10	w_{cf}	11.4	32.7	16.8	11.4	32.7	16.8
	w_{cf-idf}	<u>11.8</u>	37.3	17.9	<u>11.8</u>	37.3	17.9
	DE	8.5	28.1	13.1	08.5	28.1	13.1
	BET	9.6	33.2	14.9	09.6	33.2	14.9
	PR	11.6	<u>39.7</u>	<u>18.0</u>	11.6	<u>39.7</u>	<u>18.0</u>
	$w_{SBERT}(Ours)$	**13.9**	39.5	**20.6**	**14.0**	**39.8**	**20.7**
	Improvement (%)	17.8	-	14.5	18.6	0.25	15

Recognizing the benefits of incorporating context to enhance the accuracy of the generated concepts, we further leverage the contextual score assigned by DBpedia Spotlight, obtained as a result of the concept identification step (see Section 4.3), to increase the weights of the identified concepts. The contextual score considers the surrounding context of the text. It is calculated for each potential entity, based on various factors. These include the frequency of the entity in the DBpedia knowledge base, the context in which

it appears in the text, and the co-occurrence of other words in the text that are related to the entity [51]. We modify the weights of the concepts in the learning material to the harmonic mean of the similarity score between the learning material and the concept, and the contextual score assigned by DBpedia Spotlight, as defined in Equation 1. The weights of the related concepts and the categories remain the same as described earlier. This resulted in an enhancement of the F1-score for the selection of the top 5 and top 10 ranked concepts utilizing the w_{SBERT} strategy (see Table 4, with harmonic mean). For the top three ranked concepts, the PR strategy remains the best strategy followed by the w_{cf-idf} strategy. Overall, our evaluation results show that w_{SBERT} with the harmonic mean is the best-performing concept-weighting strategy when extracting five or more concepts in learning materials.

$$w(lm_i, c) = \frac{2 \times contextualScore(c) \times cos(emb_c, emb_{lm_i})}{contextualScore(c) + cos(emb_c, emb_{lm_i})} \quad (1)$$

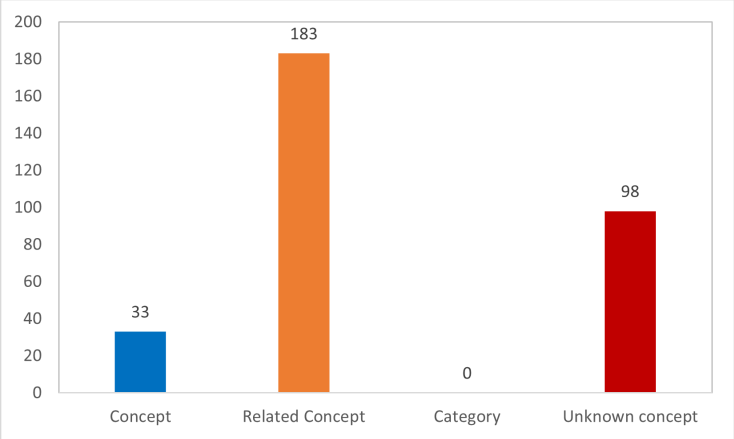

Figure 7. Annotated concept distribution in the CCI dataset, mapped to our results.

6. Conclusions and Future Work

In this work, we aimed to answer the research question "How to leverage knowledge bases and word/sentence embedding techniques to automatically construct an Educational Knowledge Graph (EduKG) based on the concepts extracted from learning materials?". To answer this, we proposed a pipeline for the automatic construction of EduKGs, in an unsupervised manner, relying on state-of-the-art word and sentence embedding techniques. Furthermore, we conducted extensive experiments on different datasets, demonstrating significant accuracy and efficient improvements over several state-of-the-art keyphrase extraction and concept-weighting strategies. Our evaluation results confirm that word and sentence embeddings provide a simple, yet powerful method to effectively and efficiently construct EduKGs in an automatic manner. As part of future research, we will aim to extensively evaluate the accuracies of the constructed EduKGs with CourseMapper users. Moreover, we plan to follow a human-in-the-loop approach to improve the quality of the automatically constructed EduKGs. Furthermore, we plan to utilize the constructed EduKGs for learning materials in CourseMapper to recommend personalized learning resources (e.g., YouTube videos and Wikipedia articles), as well as related concepts, based on the learners' current knowledge state.

Author Contributions: Conceptualization, Q.U.A. and M.A.C.; methodology, Q.U.A., M.A.C. and K.G.C.B.; software, K.G.C.B. and S.J.; validation, M.A.C.; writing—original draft preparation, Q.U.A. and K.G.C.B.; writing—review and editing, Q.U.A. and M.A.C.; visualization, Q.U.A., S.J., K.G.C.B. and R.A.; supervision, M.A.C., S.J. and Q.U.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: We acknowledge the support from the Open Access Publication Fund of the University of Duisburg-Essen.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; Zhang, W. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 601–610.
2. Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; Melo, G.D.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge graphs. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–37. [CrossRef]
3. Chen, P.; Lu, Y.; Zheng, V.W.; Chen, X.; Yang, B. Knowedu: A system to construct knowledge graph for education. *IEEE Access* **2018**, *6*, 31553–31563. [CrossRef]
4. Novak, J.D.; Cañas, A.J. The theory underlying concept maps and how to construct and use them. *Fla. Inst. Hum. Mach. Cogn.* **2008**, *1*, 1–31.
5. Netflix. Available online: https://www.netflix.com/de/ (accessed on 3 April 2023).
6. Apple Siri. 2017. Available online: https://www.apple.com/siri/ (accessed on 23 April 2023).
7. IBM Watson. 2017. Available online: https://www.ibm.com/watson/ (accessed on 30 April 2023).
8. Wolfram Alpha. Available online: https://www.wolframalpha.com/ (accessed on 30 April 2023).
9. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [CrossRef] [PubMed]
10. Yang, Y.; Liu, H.; Carbonell, J.; Ma, W. Concept graph learning from educational data. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, Shanghai, China, 2–6 February 2015; pp. 159–168.
11. Shukla, H.; Kakkar, M. Keyword extraction from Educational Video transcripts using NLP techniques. In Proceedings of the 2016 6th International Conference—Cloud System and Big Data Engineering (Confluence), Noida, India, 14–15 January 2016; pp. 105–108. [CrossRef]
12. Wang, S. *Knowledge Graph Creation from Structure Knowledge*; The Pennsylvania State University: State College, PA, USA, 2017.
13. Sun, Y.; Qiu, H.; Zheng, Y.; Wang, Z.; Zhang, C. SIFRank: A New Baseline for Unsupervised Keyphrase Extraction Based on Pre-Trained Language Model. *IEEE Access* **2020**, *8*, 10896–10906. [CrossRef]
14. Iandola, F.N.; Shaw, A.E.; Krishna, R.; Keutzer, K.W. SqueezeBERT: What can computer vision teach NLP about efficient neural networks? *arXiv* **2020**, arXiv:2006.11316.
15. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
16. Liu, H.; Ma, W.; Yang, Y.; Carbonell, J. Learning concept graphs from online educational data. *J. Artif. Intell. Res.* **2016**, *55*, 1059–1090. [CrossRef]
17. Shen, Y.; Chen, Z.; Cheng, G.; Qu, Y. CKGG: A Chinese knowledge graph for high-school geography education and beyond. In Proceedings of the Semantic Web—ISWC 2021: 20th International Semantic Web Conference, ISWC 2021, Virtual Event, 24–28 October 2021; Proceedings 20; Springer: Berlin/Heidelberg, Germany, 2021; pp. 429–445.
18. Chi, Y.; Qin, Y.; Song, R.; Xu, H. Knowledge graph in smart education: A case study of entrepreneurship scientific publication management. *Sustainability* **2018**, *10*, 995. [CrossRef]
19. Yang, X.; Tan, L. The Construction of Accurate Recommendation Model of Learning Resources of Knowledge Graph under Deep Learning. *Sci. Program.* **2022**, *2022*, 1010122. [CrossRef]
20. Chen, Q.; Xia, J.; Feng, J.; Tong, M. Research on Knowledge Graph Construction for Python Programming Language. In Proceedings of the International Conference on Smart Learning Environments, Hangzhou, China, 18–20 August 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 119–126.
21. Hubert, N.; Brun, A.; Monticolo, D. New Ontology and Knowledge Graph for University Curriculum Recommendation. In Proceedings of the ISWC 2022—The 21st International Semantic Web Conference, Virtual Event, 23–27 October 2022.
22. Morsi, R.; Ibrahim, W.; Williams, F. Concept maps: Development and validation of engineering curricula. In Proceedings of the 2007 37th Annual Frontiers in Education Conference-Global Engineering: Knowledge without Borders, Opportunities without Passports, Milwaukee, WI, USA, 10–13 October 2007; IEEE: Piscataway, NJ, USA, 2007; pp. T3H-3–T3H-6.
23. Zhu, P.; Zhong, W.; Yao, X. Auto-Construction of Course Knowledge Graph based on Course Knowledge. *Int. J. Perform. Eng.* **2019**, *15*, 2228.
24. Yang, Z.; Wang, Y.; Gan, J.; Li, H.; Lei, N. Design and research of intelligent question-answering (Q&A) system based on high school course knowledge graph. *Mob. Netw. Appl.* **2021**, *26*, 1884–1890.
25. Manrique, R.; Grévisse, C.; Mariño, O.; Rothkugel, S. Knowledge graph-based core concept identification in learning resources. In Proceedings of the Joint International Semantic Technology Conference, Awaji, Japan, 26–28 November 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 36–51.

26. Zheng, Y.; Liu, R.; Hou, J. The construction of high educational knowledge graph based on MOOC. In Proceedings of the 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 260–263.
27. Deng, Y.; Lu, D.; Huang, D.; Chung, C.J.; Lin, F. Knowledge graph based learning guidance for cybersecurity hands-on labs. In Proceedings of the ACM Conference on Global Computing Education, Chengdu, China, 17–19 May 2019; pp. 194–200.
28. Rahdari, B.; Brusilovsky, P.; Thaker, K.; Barria-Pineda, J. Using knowledge graph for explainable recommendation of external content in electronic textbooks. In Proceedings of the iTextbooks@ AIED, Virtual, 6–9 July 2020.
29. Qiao, L.; Yin, C.; Chen, H.; Sun, H.; Rong, W.; Xiong, Z. Automated Construction of Course Knowledge Graph Based on China MOOC Platform. In Proceedings of the 2019 IEEE International Conference on Engineering, Technology and Education (TALE), Yogyakarta, Indonesia, 10–13 December 2019; pp. 1–7. [CrossRef]
30. Chen, H.; Yin, C.; Fan, X.; Qiao, L.; Rong, W.; Zhang, X. Learning path recommendation for MOOC platforms based on a knowledge graph. In Proceedings of the International Conference on Knowledge Science, Engineering and Management, Tokyo, Japan, 14–16 August 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 600–611.
31. Su, Y.; Zhang, Y. Automatic construction of subject knowledge graph based on educational big data. In Proceedings of the 3rd International Conference on Big Data and Education, Chengdu, China, 21–23 August 2020; pp. 30–36.
32. Dang, F.R.; Tang, J.T.; Pang, K.Y.; Wang, T.; Li, S.S.; Li, X. Constructing an Educational Knowledge Graph with Concepts Linked to Wikipedia. *J. Comput. Sci. Technol.* **2021**, *36*, 1200–1211. [CrossRef]
33. Rahdari, B.; Brusilovsky, P. Building a Knowledge Graph for Recommending Experts. In Proceedings of the DI2KG@ KDD, Anchorage, Alaska, 5 August 2019.
34. Mondal, I.; Hou, Y.; Jochim, C. End-to-end construction of NLP knowledge graph. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online Event, 1–6 August 2021; pp. 1885–1895.
35. Stewart, M.; Liu, W. Seq2kg: An end-to-end neural model for domain agnostic knowledge graph (not text graph) construction from text. In Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, Rhodes, Greece, 12–18 September 2020; Volume 17, pp. 748–757.
36. Chen, X.; Zhang, N.; Xie, X.; Deng, S.; Yao, Y.; Tan, C.; Huang, F.; Si, L.; Chen, H. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 2778–2788.
37. Wang, T.; Li, H. Coreference resolution improves educational knowledge graph construction. In Proceedings of the 2020 IEEE International Conference on Knowledge Graph (ICKG), Nanjing, China, 9–11 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 629–634.
38. Qin, Y.; Cao, H.; Xue, L. Research and Application of Knowledge Graph in Teaching: Take the database course as an example. *Proc. J. Phys. Conf. Ser.* **2020**, *1607*, 012127. [CrossRef]
39. Grévisse, C.; Manrique, R.; Mariño, O.; Rothkugel, S. Knowledge graph-based teacher support for learning material authoring. In Proceedings of the Colombian Conference on Computing, Cartagena, Colombia, 26–28 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 177–191.
40. Chau, H.; Labutov, I.; Thaker, K.; He, D.; Brusilovsky, P. Automatic concept extraction for domain and student modeling in adaptive textbooks. *Int. J. Artif. Intell. Educ.* **2021**, *31*, 820–846. [CrossRef]
41. Zhao, B.; Sun, J.; Xu, B.; Lu, X.; Li, Y.; Yu, J.; Liu, M.; Zhang, T.; Chen, Q.; Li, H.; et al. EDUKG: A Heterogeneous Sustainable K-12 Educational Knowledge Graph. *arXiv* **2022**, arXiv:2210.12228.
42. Zhang, N.; Xu, X.; Tao, L.; Yu, H.; Ye, H.; Qiao, S.; Xie, X.; Chen, X.; Li, Z.; Li, L.; et al. Deepke: A deep learning based knowledge extraction toolkit for knowledge base population. *arXiv* **2022**, arXiv:2201.03335.
43. Sultan, M.A.; Bethard, S.; Sumner, T. Towards automatic identification of core concepts in educational resources. In Proceedings of the IEEE/ACM Joint Conference on Digital Libraries, London, UK, 8–12 September 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 379–388.
44. Wang, X.; Feng, W.; Tang, J.; Zhong, Q. Course concept extraction in MOOC via explicit/implicit representation. In Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, China, 18–21 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 339–345.
45. Ain, Q.U.; Chatti, M.A.; Joarder, S.; Nassif, I.; Wobiwo Teda, B.S.; Guesmi, M.; Alatrash, R. Learning Channels to Support Interaction and Collaboration in CourseMapper. In Proceedings of the 14th International Conference on Education Technology and Computers, Barcelona, Spain, 28–30 October 2022; pp. 252–260.
46. Bonwell, C.C. Using active learning to enhance lectures. *Appl. Econ. Perspect. Policy* **1999**, *21*, 542–550. [CrossRef]
47. Ramakrishnan, C.; Patnia, A.; Hovy, E.; Burns, G.A. Layout-aware text extraction from full-text PDF of scientific articles. *Source Code Biol. Med.* **2012**, *7*, 1–10. [CrossRef]
48. Shinyama, Y. PDFMiner—Python PDF Parser. 2007. Available online: https://unixuser.org/~euske/python/pdfminer/ (accessed on 23 April 2023).
49. ELMo. Available online: https://allenai.org/allennlp/software/elmo (accessed on 23 April 2023).
50. Arora, S.; Liang, Y.; Ma, T. A simple but tough-to-beat baseline for sentence embeddings. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.

51. Mendes, P.N.; Jakob, M.; García-Silva, A.; Bizer, C. DBpedia spotlight: Shedding light on the web of documents. In Proceedings of the 7th International Conference on Semantic Systems, Graz, Austria, 7–9 September 2011; pp. 1–8.
52. Manrique, R.; Marino, O. Knowledge Graph-based Weighting Strategies for a Scholarly Paper Recommendation Scenario. In Proceedings of the KaRS@ RecSys, Vancouver, BC, Canada, 7 October 2018; pp. 5–8.
53. Manrique, R.; Herazo, O.; Mariño, O. Exploring the use of linked open data for user research interest modeling. In Proceedings of the Colombian Conference on Computing, Cali, Colombia, 19–22 September 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 3–16.
54. Hassan, H.A.M. Personalized research paper recommendation using deep learning. In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, Bratislava, Slovakia, 9–12 July 2017; pp. 327–330.
55. Hassan, H.A.M.; Sansonetti, G.; Gasparetti, F.; Micarelli, A.; Beel, J. Bert, elmo, use and infersent sentence encoders: The panacea for research-paper recommendation? In Proceedings of the RecSys (Late-Breaking Results), Copenhagen, Denmark, 16–20 September 2019; pp. 6–10.
56. Hulth, A. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Sapporo, Japan, 11–12 July 2003; pp. 216–223.
57. Augenstein, I.; Das, M.; Riedel, S.; Vikraman, L.; McCallum, A. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv* **2017**, arXiv:1704.02853.
58. Wan, X.; Xiao, J. Single document keyphrase extraction using neighborhood knowledge. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, Chicago, IL, USA, 13–17 July 2008; Volume 8, pp. 855–860.
59. Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.M.; Nunes, C.; Jatowt, A. YAKE! Keyword extraction from single documents using multiple local features. *Inf. Sci.* **2020**, *509*, 257–289. [CrossRef]
60. Rose, S.; Engel, D.; Cramer, N.; Cowley, W. Automatic Keyword Extraction from Individual Documents. In *Text Mining: Applications and Theory*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2010; pp. 1–20. [CrossRef]
61. Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* **1988**, *24*, 513–523. [CrossRef]
62. Boudin, F. Unsupervised Keyphrase Extraction with Multipartite Graphs. In Proceedings of the NAACL, New Orleans, LA, 26–30 June 2018; pp. 667–672. [CrossRef]
63. Jardine, J.G.; Teufel, S. Topical PageRank: A Model of Scientific Expertise for Bibliographic Search. In Proceedings of the EACL, Gothenburg, Sweden, 26–30 April 2014.
64. Bougouin, A.; Boudin, F.; Daille, B. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. In Proceedings of the IJCNLP, Nagoya, Japan, 14–19 October 2013.
65. Florescu, C.; Caragea, C. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 1105–1115. [CrossRef]
66. Wan, X.; Xiao, J. CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction. In Proceedings of the COLING, Manchester, UK, 18–22 August 2008.
67. Mihalcea, R.; Tarau, P. TextRank: Bringing Order into Text. In Proceedings of the EMNLP, Barcelona, Spain, 25–26 July 2004.
68. Bennani-Smires, K.; Musat, C.; Hossmann, A.; Baeriswyl, M.; Jaggi, M. Simple Unsupervised Keyphrase Extraction Using Sentence Embeddings. *arXiv* **2018**, arXiv:1801.04470.
69. Roy, D.; Sarkar, S.; Ghose, S. Automatic Extraction of Pedagogic Metadata from Learning Content. *Int. J. Artif. Intell. Educ.* **2008**, *18*, 97–118.
70. Boudin, F. A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction. In Proceedings of the Sixth International Joint Conference on Natural Language Processing, Nagoya, Japan, 14–19 October 2013; Asian Federation of Natural Language Processing: Nagoya, Japan, 2013; pp. 834–838.