future internet

MDPI

Article

Evaluating Embeddings from Pre-Trained Language Models and Knowledge Graphs for Educational Content Recommendation

Xiu Li, Aron Henriksson *, Martin Duneld, Jalal Nouri and Yongchao Wu

Department of Computer and Systems Sciences, Stockholm University, NOD-Huset, Borgarfjordsgatan 12, 16455 Stockholm, Sweden; xiu.li@dsv.su.se (X.L.); xmartin@dsv.su.se (M.D.); jalal@dsv.su.se (J.N.); yongchao.wu@dsv.su.se (Y.W.)

* Correspondence: aronhen@dsv.su.se

Abstract: Educational content recommendation is a cornerstone of AI-enhanced learning. In particular, to facilitate navigating the diverse learning resources available on learning platforms, methods are needed for automatically linking learning materials, e.g., in order to recommend textbook content based on exercises. Such methods are typically based on semantic textual similarity (STS) and the use of embeddings for text representation. However, it remains unclear what types of embeddings should be used for this task. In this study, we carry out an extensive empirical evaluation of embeddings derived from three different types of models: (i) static embeddings trained using a concept-based knowledge graph, (ii) contextual embeddings from a pre-trained language model, and (iii) contextual embeddings from a large language model (LLM). In addition to evaluating the models individually, various ensembles are explored based on different strategies for combining two models in an early vs. late fusion fashion. The evaluation is carried out using digital textbooks in Swedish for three different subjects and two types of exercises. The results show that using contextual embeddings from an LLM leads to superior performance compared to the other models, and that there is no significant improvement when combining these with static embeddings trained using a knowledge graph. When using embeddings derived from a smaller language model, however, it helps to combine them with knowledge graph embeddings. The performance of the best-performing model is high for both types of exercises, resulting in a mean Recall@3 of 0.96 and 0.95 and a mean MRR of 0.87 and 0.86 for quizzes and study questions, respectively, demonstrating the feasibility of using STS based on text embeddings for educational content recommendation. The ability to link digital learning materials in an unsupervised manner—relying only on readily available pre-trained models—facilitates the development of AI-enhanced learning.

Keywords: educational content recommendation; AI-enhanced learning; pre-trained language models; ensemble embeddings; knowledge graph embeddings; text similarity; textual semantic search; natural language processing

check for updates

Citation: Li, X.; Henriksson, A.; Duneld, M.; Nouri, J.; Wu, Y. Evaluating Embeddings from Pre-Trained Language Models and Knowledge Graphs for Educational Content Recommendation. *Future Internet* **2024**, *16*, 12. https://doi.org/10.3390/fi16010012

Academic Editor: María N. Moreno García, Fernando De la Prieta Pintado

Received: 1 December 2023
Revised: 24 December 2023
Accepted: 25 December 2023
Published: 29 December 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The digitalization of education has significantly transformed the learning landscape, resulting in the availability of extensive and diverse learning resources. These digital resources encompass a wide array of formats, including textbooks, exercises, lecture slides, video lectures, and supplementary materials both within and outside dedicated learning platforms. The sheer volume and diversity of materials can be overwhelming, making it challenging for learners to identify the most relevant and beneficial content for their needs. To that end, educational recommendation systems play a crucial role in enhancing the accessibility of relevant content, user engagement, and the effectiveness of learning resource usage by filtering contents, finding items of interest, and suggesting useful learning resources that can meet curriculum goals, users' needs, and profiles. There have been several efforts to develop educational recommendation systems for different purposes, for

instance, to provide personalized recommendations of remedial learning materials [1] or to provide a personalized review module [2], while others have proposed different solutions for recommending external learning material in the form of Wikipedia pages and YouTube videos [3–5]. In this study, we focus on educational content recommendations based on textbook content and exercises. Textbooks remain foundational resources in education, offering structured and comprehensive coverage of academic subjects, while exercises provide formative assessment and facilitate self-regulated learning [6]. Interlinking exercises and relevant textbook content allow learners to practice and reinforce newly acquired knowledge immediately after studying a particular concept. Similarly, it can facilitate the recommendation of supportive learning resources related to an exercise, providing a reference to the question and connecting this to the textbook sections where the examined knowledge points reside.

There are different approaches to developing content recommendation systems; these are summarized in Section 2.1. The most common approach is based on semantic textual similarity (STS), essentially allowing for calculating the semantic similarity between two texts. This technique forms the backbone of an educational content recommender and allows it to automatically link and recommend learning materials. The key to effective and accurate STS-based content recommendation systems is powerful text representations. The overarching aim of our study is therefore to evaluate different text representations and identify which ones to use in an STS-based educational content recommendation system based on automatically interlinking textbook content and exercises. On the one hand, transformer-based models, such as BERT [7] and GPT-4 [8], consistently achieve state-of-the-art performance on a variety of tasks and continue to dominate the field of natural language processing (NLP). While these pre-trained language models have acquired a profound understanding of intricate patterns and higher-order co-occurrence statistics in text data [9], they tend to struggle with capturing domain-specific, factual knowledge [10,11]. On the other hand, knowledge graphs play an important role in a variety of knowledge-driven applications and AI tasks [12], such as recommendation systems [13,14] and information retrieval systems [15], and are excellent at representing structured knowledge.

In a previous study [16], we investigated a number of different text representations and ways of calculating semantic similarity between textbook content and quizzes. In terms of text representations, traditional statistical models such as TF–IDF (Term Frequency–Inverse Document Frequency) were compared to Doc2Vec [17], BERT, and ConceptNet-based models. Paragraph-level embeddings were employed for representing textbook content, and three different pooling methods were investigated to represent a given textbook section. The results showed that using a pre-trained language model in SBERT outperformed the other models, while max paragraph pooling outperformed mean pooling and adjusted mean pooling. The latter result can be viewed as indicating—at least for the purpose of identifying textbook sections that are relevant for a given quiz—that high semantic similarity with a single paragraph in a section is more important than high average semantic similarity across all paragraphs in a section.

In this study, we carry out a thorough investigation and evaluation of text representations in the form of embeddings from pre-trained language models to be leveraged in an STS-based educational content recommendation system. In particular, we include in our evaluation the use of higher-dimensional embeddings derived from an LLM and continue to explore the potential of incorporating knowledge graphs. With respect to the latter, we do so in one of two ways: (i) by representing texts—both exercises and textbook content—based solely on pre-identified concepts, and (ii) by combining knowledge graph embeddings with contextual embeddings in various ensembles in an attempt to exploit their respective advantages. The STS-based educational content recommendation system is designed to automatically link relevant textbook content to exercises, here, extended to encompass both quizzes and study questions across three subjects. In summary, the main contributions and findings of this study are described below:

- We evaluate the use of static and contextual embeddings derived from different types of pre-trained language models and a concept-based knowledge graph for STS-based educational content recommendation. The results show that higher-dimension contextual embeddings from LLM outperform both lower-dimension contextual embeddings and static embeddings trained using a knowledge graph, even when resorting to domain-specific concept filtering.
- We explore the possibilities of combining the respective advantages of contextual embeddings and knowledge graph embeddings using three different ensemble methods based on both early and late fusion approaches. The results show that when using contextual embeddings from a relatively smaller language model and lower dimension (SBERT), performance improves when combined with knowledge graph embeddings in a late fusion fashion. However, when using higher-dimension embeddings from LLM (`text-embedding-ada-002`), there is no significant gain from any of the ensembles.
- We show that by leveraging powerful embeddings from a pre-trained language model, an STS-based educational recommendation system can be developed in a wholly unsupervised manner, i.e., without relying on labeled training data. The system can accurately link exercises to relevant textbook sections: the best-performing model obtains a Recall@3 of 0.96 and an MRR of 0.87 for quizzes and a Recall@3 of 0.95 and an MRR of 0.86 for study questions across three subjects.

The paper is structured as follows. In Section 2, we review existing literature and describe previous work relevant to our study, providing a comprehensive overview of the current state of educational content recommendation systems, knowledge graph embeddings, as well as methods for incorporating knowledge graphs with pre-trained language models. In Section 3, we first outline the datasets utilized in our research, which contain textbook instructive content and two types of exercises: quizzes and study questions. After that, we present the study design, models, evaluation metrics, and significant testing. Section 4 presents the results of our experiments and analyses. Subsequently, in Section 5, we delve into the implications of the results, contextualizing our findings within the broader research landscape. Finally, Section 6 provides a succinct summary of the study and its contributions.

2. Related Work

In this section, we first describe related work on designing and developing educational content recommendation systems, summarizing the main approaches. We then introduce the concept of knowledge graphs and describe how embeddings can be trained using the information captured by such graphs. Finally, we describe efforts to integrate knowledge graphs and pre-trained language models in order to harness the advantages that each of these models brings.

2.1. Educational Content Recommendation

Educational content recommendation has gained significant attention in recent years, with researchers exploring various methodologies to enhance learning experiences. Recommendation systems in education aim to address challenges such as learner engagement, content relevance, and adaptability. Recommendation algorithms are generally based on (i) heuristic rules, (ii) content similarity, or (iii) behavior similarity. Rule-based recommendation systems involve intricate human logic and consideration of content attributes such as difficulty level, completion time, and types, providing a pre-configured content subscription. Content similarity-based recommenders focus on the content itself, projecting the content into numeric representations within a feature space and evaluating the semantic distance between texts. The semantics of learning materials can be captured through ontology-based approaches, representing educational concepts [1,3,18], or through contextual semantic meanings of entire sentences [19,20] in an approach based on contextual semantics. Learning materials can then be linked using methods like exact concept phrase

matching or using word embeddings for STS. Recommenders based on user behavior tailor recommendations to individual students' needs, analyzing profiles based on knowledge level, preferences, and learning style. Collaborative filtering is a user behavior-based approach that groups students with similar behavior patterns and recommends resources based on collective engagement. Content recommendation systems often leverage information retrieval and semantic search technologies for content processing, integrating rule-based and behavior-based perspectives for more relevant recommendations. Niu et al. [18] combined an ontology-based content similarity approach with students' behaviors, using extracted concepts from the learning contents to construct a knowledge graph and aggregating concept similarities and students' behavior (student–concept interactions) into the knowledge graph for collaborative content recommendation. Rahdari et al. [3] used a graph database, Neo4j, to store the constructed knowledge graph that contains Wikipedia articles, textbook content, and the student model, and used Neo4j's internal full-text search engine, Lucene, to calculate the relevance scores for the recommendations. Hybrid approaches, combining multiple methods, are also prevalent [1,3,21,22].

2.2. Knowledge Graph Embeddings

A knowledge graph is a structured representation of knowledge that captures general knowledge relationships between entities in a domain, often represented as nodes and edges in a graph. Examples of large-scale knowledge graphs are Wikidata [23], YAGO [24], Freebase [25], and DBpedia [26]. Knowledge graph embeddings refer to techniques that transform entities and relationships in a knowledge graph into numerical representations in a feature space that can capture semantic relationships and patterns within the graph. When it comes to modeling methods, a variety of approaches are available. One traditional mathematical method involves representing the relational space using a term–term matrix, positive pointwise mutual information, and singular value decomposition, as demonstrated in ConceptNet [27]. Another approach is based on relation paths, representing relations linearly as translations, for instance, as exemplified by TransE [28]. Alternatively, graph neural networks can be employed, utilizing graph convolutional layers or similar architectures to aggregate information from neighboring entities, as seen in models like GraphSAGE [29] and R-GCN [30]. Random walk-based methods utilize graph traversals to generate sequences of entities, which are used to train the embeddings, as seen in Node2Vec [31], capturing both local and global structural information. Attention-based mechanisms, such as KGAT (Knowledge Graph Attention Network) [32], have also been applied. Additionally, enriching knowledge graphs with descriptive external information is another strategy, which involves incorporating supplementary details like entity type [33] or descriptive information of entities [34–36]. Knowledge graph embeddings are usually derived from diverse sources, including crowd-sourced data and hybrid methods.

2.3. Combining Knowledge Graphs and Pre-Trained Language Models

Knowledge graph embeddings excel in structured knowledge representation, while pre-trained language models are powerful for understanding and generating natural language text in various contexts. Extensive research has been conducted to harness the advantages of both structured knowledge and unstructured text comprehension. Below, we summarize strategies for combining knowledge graphs with pre-trained language models into three categories:

- **Traditional ensemble methods**. These encompass the most commonly used ensemble methods. The ensemble can be conducted directly with the embeddings, i.e., feature-level fusion, which is also called early fusion. A straightforward way to combine embedding vectors is simply to "stack" them into a higher-dimension embedding vector [37]. An alternative is to add up the embedding vectors in a linear way, for example, through a linear interpolation, where the weights can be either preset or trained [1,37,38]. Learning a projection function between embeddings is another solution. For linear projection, Muromägi et al. [39] conducted linear regression in two

ways, i.e., ordinary least squares and orthogonal procrustes to learn the transformation matrix. Gammelgaard et al. [40] instead used ridge regression and generalized procrustes analysis. The projection function can also be non-linear, for example, using a neural network to learn a multilayer perceptron (MLP) [41]. An alternative to early fusion ensembles is to perform the combination in the decision stage, i.e., late fusion. Each individual model makes their own independent decisions, which are then combined or fused to produce the final result [42].

- **Retrofitting**. Originally proposed in [43], this process uses external lexical relational resources from knowledge graphs to refine an existing matrix of word embeddings to obtain higher-quality semantic vectors. Speer et al. [27] used a generalization of the "retrofitting" method to create ensembles that learned from Word2Vec, GloVe, and ConceptNet. Faruqui et al. [43], Fang et al. [44,45] further leveraged the information in words, in relations embedded in edges and multisense knowledge, and developed extensions of the retrofit ensemble models.
- **Injecting external structured knowledge in the transformer architecture**. One method involves incorporating entity embeddings using a layerwise local/global fusion strategy [34,46]. Alternatively, Ri et al. [47] trained an entity-aware BERT model using the self-attention mechanism and masked language modeling objective. Another approach involves a deep integration through knowledge attention and a recontextualization mechanism [48]. Injecting external knowledge in the transformer architecture requires further training or fine-tuning of the language models, which means they are relatively computationally expensive methods.

3. Methods

In this section, we elucidate the dataset utilized in this study and expound upon the methodologies employed in the evaluation of different embeddings in an educational content recommender based on semantic textual similarity. We aim to investigate the potential of pre-trained language models for this task and explore the use of three different types of pre-trained embedding models, which are described below. In addition to using the models individually, we also explore ensembles of embeddings and evaluate different strategies for combining models. To evaluate the performance of the embeddings on the task of recommending textbook content based on an exercise, we use two metrics in Recall@3 and mean reciprocal rank, and employ McNemar's test for testing whether observed differences in performance are statistically significant.

3.1. Data

The data used in this study are instructive textbook content and the single/multiple-choice quizzes and study questions from three Swedish digital textbooks: Biology G7-9, Social Science G11, and Physics G7-9. The textbooks follow a hierarchical structure of Chapter→Section→Sub-section. Each section, presented as a web page in the frontend, encompasses multiple subsections and associated exercises. To obtain labeled data for evaluating our solution, we leverage the web page's structure, where exercises are situated on the same page as the corresponding section in the textbook. Hence, each exercise labels the section where it resides as a TRUE recommendation. This setup eliminates the need for additional human annotation, assuming that each exercise has only one corresponding relevant section as the TRUE recommendation in the textbook. While quizzes or study questions often focus on a limited knowledge scope that can typically be explained within a section, there might be instances where the same knowledge point spans multiple sections. Nevertheless, our auto-labeling approach suffices for evaluating the content recommender, as validated by manual inspections of sample pages and supported by our case-adjusted evaluation metrics introduced in Section 3.6.

Table 1 shows the statistics of the data, encompassing a total of 147 textbook sections, 930 quizzes, and 1625 study questions. By employing the concept summary at the end of each section, we calculate concept density, revealing that Biology and Physics exhibit a

higher richness in concepts than Social Science. Social Science is, on the other hand, exercise-intensive, with an average of 11 quizzes and 16 study questions per section. Quizzes are single or multi-choice questions. Study questions are inquiries or prompts designed to guide the process of learning. The data for study questions contain only the questions, without answers. The complexity level of study questions can range from simple recall-based questions to more analytical or calculation-related questions. Generally, answers can be found or inferred with references to the relevant textbook content. The complexity level of quizzes is somewhat higher than study questions since the questions are sometimes not that straightforward.

Table 1. Statistics of the textbook data.

Statistic	Biology	Social Science	Physics
# chapters	6	11	11
# sections	56	34	57
# sub-sections	585	288	457
# paragraphs	5834	2762	3741
avg # tokens per paragraph	22	41	18
Concept density: avg # tokens per concept	**183**	3643	**184**
# quizzes	264	386	280
avg # quizzes per section	5	**11**	5
# study questions	550	556	519
avg # study questions per section	10	**16**	9

3.2. Pre-Trained Embedding Models

In this study, we consider three different types of pre-trained embedding models: (i) contextual embeddings from a pre-trained language model, (ii) contextual embeddings from an LLM, and (iii) static embeddings trained using a concept-based knowledge graph. Figure 1 illustrates how these models are used individually.

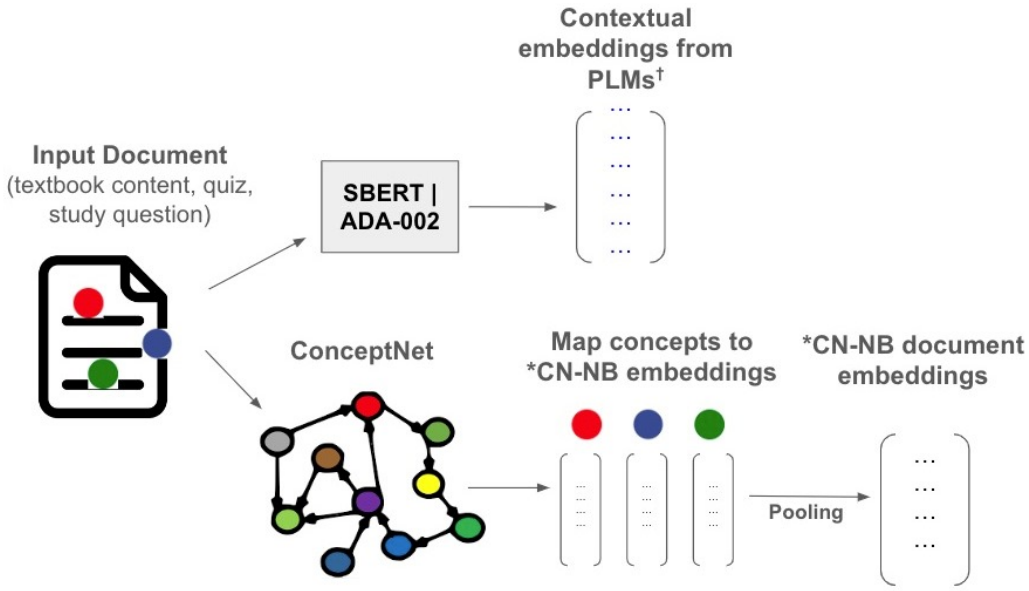


Figure 1. Illustration of how models are used individually. CN-NB * = ConceptNet Numberbatch. PLM † = Pre-trained Language Model.

3.2.1. Contextual Embeddings from a Pre-Trained Language Model

Unlike traditional language models that treat words in isolation, contextual models offer a nuanced understanding of linguistic context. A pioneering Open Source exemplar in this realm is BERT (Bidirectional Encoder Representations from Transformers). BERT

is a transformer-based model that uses deep bidirectional architectures [7]. KB-BERT is a pre-trained Swedish BERT model [49]. In this study, we use SBERT [50], which is sentence KB-BERT, an extension of the KB-BERT model. In our previous study [16], SBERT proved to perform better than traditional models and ConceptNet Numberbatch models for educational content recommendations. In addition, the benchmark results on translated STS-B data show that KB-BERT outperforms other Swedish semantic representation models on STS tasks [51].

3.2.2. Contextual Embeddings from a Pre-Trained Large Language Model

GPT (Generative Pre-trained Transformer) is an autoregressive and unidirectional transformer model [52]. Although GPT models were originally designed for text generation tasks, GPT-based embedding models have also been used for text similarity-related tasks (https://platform.openai.com/docs/guides/embeddings. Last accessed 28 November 2023) [53]. `text-embedding-ada-002` (henceforth referred to as ADA-002) is OpenAI's second-generation embeddings-as-a-service API endpoint model specifically adapted for text embeddings and is multilingual (including Swedish). ADA-002 is recommended by OpenAI for text similarity tasks since it outperforms the previously most capable first-generation embedding model `text-similarity-davinci-001` on most text similarity-related tasks and is also cheaper and computationally lighter [54]. The first-generation embedding models of OpenAI are reported to be GPT-based (https://platform.openai.com/docs/guides/embeddings. Last accessed 28 November 2023) [53]. However, as a proprietary commercial product, no detailed information, such as architecture, training data, or model, has been disclosed publicly for ADA-002 since its release in late 2022. With the recent popularization of LLMs and embeddings-as-a-service, we include ADA-002 for semantic similarity in this study. ADA-002 embeddings have 1536 dimensions [54], and we assume these have been derived from an LLM.

We have also explored the possibility of using GPT-SW3—AI Sweden's recent release, which is multilingual but focuses on the Nordic languages [55]. It is the first truly large-scale generative language model for the Swedish language and is based on the same technical principles as the much-discussed GPT-4 (https://www.ai.se/en/project/gpt-sw3. Last accessed 28 November 2023). The current status of GPT-SW3 is not suitable for sentence embeddings since it is not especially further trained for embeddings (https://lab.kb.se/leaderboard/results. Last accessed 28 November 2023). Yet, it provides possibilities for a Swedish GPT-based text embedding solution, which is Open Source in contrast to OpenAI.

3.2.3. Static Embeddings from a Concept-Based Knowledge Graph

Compared to contextual embeddings derived from pre-trained language models, the strength of knowledge graph models lies in their ability to represent structured information and explicit relationships. The integration of knowledge graphs in AI systems may enhance the performance and explainability of the system [12]. There is a wide range of pre-trained knowledge graph models, such as those mentioned in Section 2.2; however, we find only two options for the Swedish language: ConceptNet Numberbatch and mLUKE. Our further examination shows that mLUKE has only 30,129 Swedish entities [47], while Numberbatch has 167,321 entities (https://github.com/commonsense/conceptnet-numberbatch. Last accessed 28 November 2023) [27].

ConceptNet is a knowledge graph that is designed to capture common-sense knowledge and relationships between concepts [27]. It calculates word embeddings directly from a sparse, symmetric term–term matrix that represents the ConceptNet graph, using positive pointwise mutual information and truncated singular value decomposition [27]. ConceptNet Numberbatch is a pre-trained ConceptNet extention model. It is an ensemble of models that combines knowledge and semantic properties from ConceptNet, Word2Vec, and GloVe, using a variation on retrofitting [27]. Therefore, it is a ConceptNet-based ensemble model.

3.3. Semantic Textual Similarity

Given two input texts, we want to derive a score that measures their similarity at a semantic level, rather than their simple lexical features [56]. Specifically, in our study, we define each exercise as q_i from the query set $\{Q : q_1, \ldots, q_i, \ldots, q_n\}$ to conduct the similarity search in the textbook corpus $\{C : \{p_1 : c_1^1, \ldots, c_{n_1}^1\}, \ldots, \{p_k : c_1^k, \ldots, c_j^k, \ldots, c_{n_k}^k\}, \ldots, \{p_P : c_1^P, \ldots, c_{n_P}^P\}\}$, where P is the number of web pages (sections) in the corpus, n_k is the number of paragraphs on page k, v_i is the embedding vector of query q_i, u_j^k is the embedding vector of the jth paragraph on page k.

We employ cosine similarity to measure semantic textual similarity between q_i and page k:

$$Sim(v_i, u^k) = \frac{v_i \cdot u^k}{\mid v_i \mid\mid u^k \mid}, \tag{1}$$

where u^k is the embedding vector of page k.

3.4. Pooling Methods

In our previous study [16], max-pooling was shown to surpass mean-pooling and adjusted-mean pooling in both Recall@3 and MRR. We therefore adopt max-pooling as the paragraph-level pooling strategy in this study.

$$\text{Max-pooling:} \quad \max_{1 \le j \le n_k} Sim(v_i, u_j^k) \tag{2}$$

ConceptNet Numberbatch also needs a word-level pooling method to be defined since the model is concept-based. When there is more than one identified concept according to the ConceptNet vocabulary, we use mean-pooling as the concept-level pooling method, following [57]. When no concept can be identified in the text, we set the embedding vector to be a zero vector.

For ConceptNet Numberbatch, it is necessary to define a word-level pooling method since the model is concept-based. When multiple concepts are identified based on the ConceptNet vocabulary, mean-pooling is employed as the concept-level pooling method, as described in [57]. When no concept is identified, the embedding vector is set to a zero vector [58].

3.5. Ensemble Methods

In the context of NLP, ensembles enable the fusion of distinct semantic representations for a more comprehensive and multifaceted representation of language. It has, moreover, been shown that the utilization of ensemble models can bring advantages such as increased robustness, improved generalization, and overall superior performance compared to the deployment of individual models [10]. We want to combine the power of state-of-the-art transformer models with a knowledge graph-based model through different ensemble strategies. We employ commonly used ensemble methods due to not requiring any further pre-training or fine-tuning. In addition, certain linear ensemble methods allow us to further examine the contribution of contextual embeddings and knowledge graph-based embeddings when they are fused.

3.5.1. Learning a Projection Function

We train and learn a projection function on the intersection words of the pre-trained embedding models (SBERT and ADA-002) and ConceptNet Numberbatch, denoted by $\Rightarrow$. The training direction is from SBERT or ADA-002 feature spaces to the ConceptNet Numberbatch space. This is because the pre-trained language models such as SBERT and ADA-002 tend to contain more words than ConceptNet Numberbatch since they are trained on large datasets and use WordPiece tokenization.

Given $\{E : e_1, \ldots, e_i, \ldots, e_{167321}\}$ as the concept vocabulary of ConceptNet Numberbatch, $\{Y : y_1, \ldots, y_i, \ldots, y_{167321}\}$ is the corresponding ConceptNet Numberbatch embedding vector set, and $\{W : w_1, \ldots, w_i, \ldots, w_{167321}\}$ is the corresponding concept entity embeddings from the pre-trained contextual language models, i.e., SBERT or ADA-002, respectively, in our case. A function f for $f(W) \approx Y$ is trained. Here, we represent the function as a multi-layer perceptron (MLP). Starting from initial random weights, MLP minimizes the loss function by repeatedly updating these weights. After computing the loss, a backward pass propagates it from the output layer to the previous layers, providing each weight parameter with an update value meant to decrease the loss. We set the MLP to be two hidden layers and train it with the sum of two loss functions: mean squared error (MSE) + mean cosine similarity (MCS). Other parameters are set: epochs = 50, batch_size = 32, validation_split = 0.2, optimizer = adam. The MLP is trained on the intersecting words of SBERT | ADA-002 embeddings and ConceptNet Numberbatch. After training the MLP, the remaining word embeddings that exist in SBERT | ADA-002 but not ConceptNet Numberbatch are projected into the ConceptNet Numberbatch space. Note that we project from high-dimensional spaces of pre-trained language models to the 300-dimension ConceptNet Numberbatch space, as SBERT and ADA-002 embeddings have 768 and 1536 dimensions, respectively. Figure 2 provides an overview of the proposed method.

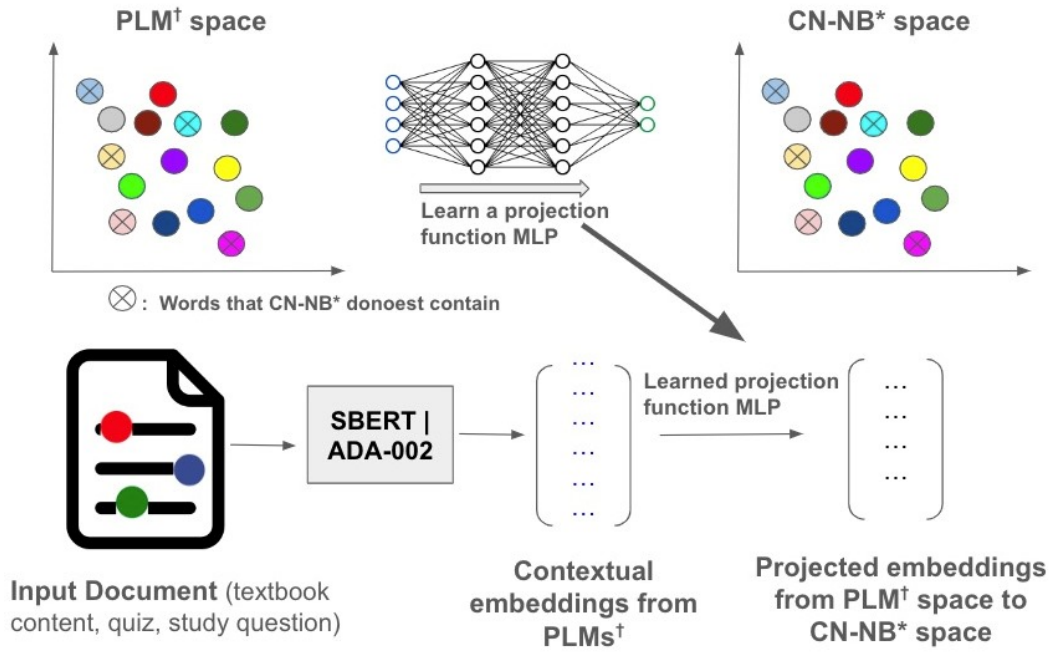


Figure 2. Overview of ensemble method based on projecting contextual embeddings of pre-trained language models into the ConceptNet Numberbatch space. CN-NB * = ConceptNet Numberbatch. PLM † = Pre-trained Language Model.

3.5.2. Stacking Embeddings

Stacking the embeddings' vectors from different models simply means concatenating the vectors vertically to create a combined higher-dimension embedding vector. This is another early fusion ensemble method. We denote this method using $\bigoplus$. For the input text, we first retrieve sentence embeddings from SBERT | ADA-002. For sentence embeddings from ConceptNet Numberbatch, we follow the method explained in Section 3.4. We then stack the two sentence embedding vectors. The dimension of the stacked embeddings is equal to the sum of the two embedding vectors' dimensions. Figure 3 provides an overview of the proposed method.

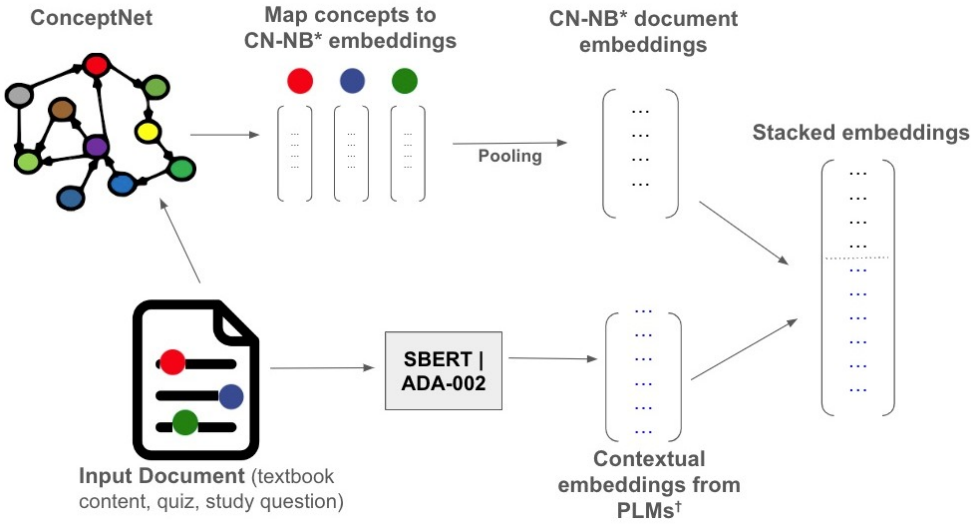


Figure 3. Overview of ensemble method based on stacking the embeddings from two models. CN-NB * = ConceptNet Numberbatch. PLM † = Pre-trained Language Model.

3.5.3. Linear Interpolation of Similarity Scores

Fusing the similarity scores of two models occurs after each model has measured the semantic textual similarity of each text pair. Hence, this is a late fusion ensemble method. A linear interpolation is typically a weighted average where different weights are applied to each model's embeddings (see Equation (3)). The weights can be learned during training or assigned based on knowledge of the performance of individual models. We denote this method using $+\!\!+$.

$$Sim_{q_i,t_k}^{fused} = \alpha \cdot Sim_{q_i,t_k}^{PLM} + (1-\alpha) \cdot Sim_{q_i,t_k}^{CN-NB}, \tag{3}$$

where Sim_{q_i,t_k} is the similarity between question q_i and section t_k. The default α is set to be 0.5, which means that pre-trained language model embeddings and ConceptNet Numberbatch embeddings have equal weight in the fusion. In Section 4.3, we present the linear interpolation results in detail when the alpha is set between 0 and 1 with the gradient 0.1. Figure 4 provides an overview of the proposed method.

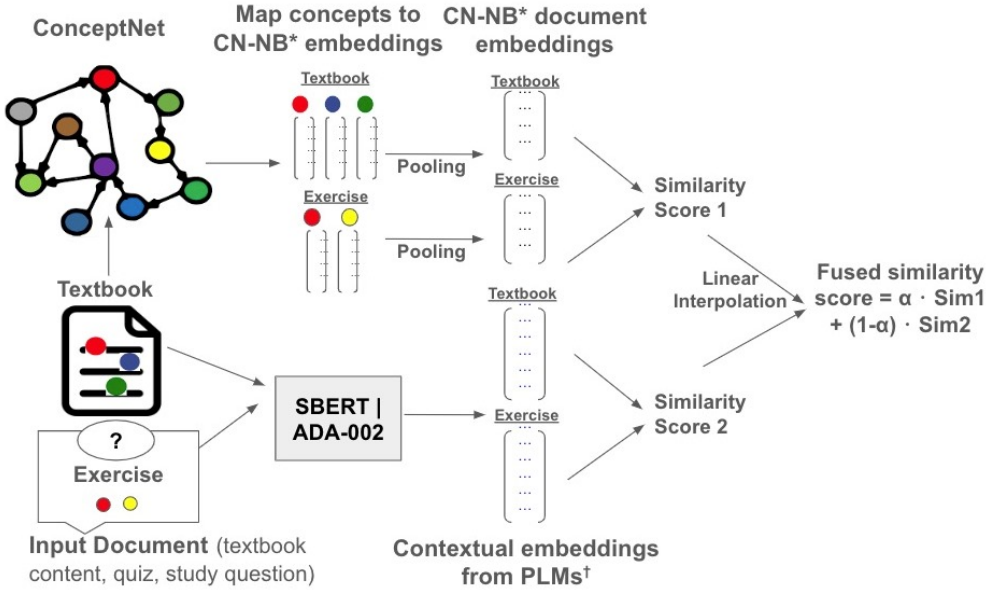


Figure 4. Overview of ensemble method based on fusing the similarity scores through a linear interpolation. CN-NB * = ConceptNet Numberbatch. PLM † = Pre-trained Language Model.

3.6. Evaluation Metrics

We request the models to recommend three relevant sections for each exercise, and the top three sections with the highest cosine similarity scores will be recommended. We adopt Recall@3 and Mean Reciprocal Rank (MRR) to evaluate the results of our content recommendation systems.

- Recall@3 is a modified version of the original recall metric, which measures binary output. Top-N recall is also called high Hits@N. For each query, if the recommended top three sections include our TRUE-labeled section, it returns 1, otherwise returns 0. The Recall@3 score is then obtained by averaging the Boolean values across all n queries. (See Equations (4) and (5).)

$$RecBoolean_i = \begin{cases} 1 & \text{if } p_i^{\text{True}} \in Top@3(p_1^{\text{rec}}, p_2^{\text{rec}}, p_3^{\text{rec}}) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$Recall@3 = \frac{1}{n} \cdot \sum_{i=1}^{n} RecBoolean_i \tag{5}$$

- Mean Reciprocal Rank (MRR) calculates the average of the inverse of the ranks where the first relevant section was retrieved for a set of queries. It assesses the model's effectiveness in recommending the TRUE relevant section towards the top of the ranking as in Equation (6).

$$MRR = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{1}{rank_i}, \tag{6}$$

where $rank_i$ is the rank position of the TRUE relevant page for the ith query.

3.7. Significance Testing

In order to see if the observed differences between the evaluated models is statistically significant or occurred due to chance, we perform significance testing. More specifically, we employ McNemar's test, which is also referred to as "within-subjects chi-squared test". It is a statistical test that is used when comparing two models or interventions on binary outcomes. The test analyzes paired dichotomous measurements in a 2×2 contingency table and assesses whether there is a significant difference in the marginal frequencies of two related variables. In machine learning, McNemar's test can be employed to evaluate and compare the predictions of two models to each other by examining the instances where their predictions differ. By comparing the number of discordant pairs (cases where one model is correct while the other is incorrect), McNemar's test helps to identify whether the observed differences are statistically significant.

In our experiments, we apply McNemar's test to compare the performance of selected pairs of models. Similar to Recall@3, for each query, if the recommended top three sections include our TRUE-labeled section, it returns 1, otherwise it returns 0. The 2×2 contingency table is calculated as in Table 2, with 1 representing OK and 0 representing FAIL.

Here, the null hypothesis for McNemar's test is that the content recommender performs equally well to hit the True-labeled section in the top three recommendations with model 1 and model 2 (i.e., there is no difference between the outcomes of the two models). The alternative hypothesis, then, is that there is a statistical difference in the performance of the two tested models' outcomes. The McNemar test statistic ("chi-squared") is computed as follows:

$$\chi^2 = \frac{(B - C)^2}{B + C} \tag{7}$$

A significance level of $\alpha = 0.05$ was used for hypothesis testing. This means that if McNemar's test gives a p-value below 0.05, we have less than a 5% probability of erroneously rejecting the null hypothesis.

Table 2. The contingency table in McNemar's test.

	Model 1 OK	Model 1 FAIL
Model 2 OK	**A**	**B**
Model 2 FAIL	**C**	**D**

4. Results

In this section, we first present the individual results using embeddings from three types of models, i.e., contextual embeddings from two pre-trained language models (SBERT and ADA-002) and static embeddings trained using a concept-based knowledge graph (ConceptNet Numberbatch). Following that, three ensemble methods are evaluated that combine knowledge graph-based embeddings and contextual embeddings. We further analyze the results from the late-fusion ensemble model using linear interpolation. We also report results from exploring the use of ConceptNet Numberbatch with domain-specific concept filtering. Finally, we describe potential differences in performance across subjects and types of exercises.

4.1. Evaluation of Single Models

Table 3 shows the results of using embeddings derived from a single model. As can be seen, ADA-002 demonstrates superior performance compared to SBERT with respect to both Recall@3 and MRR, in all subjects and both exercise types, with the sole exception being biology quizzes for Recall@3. The McNemar test shows that Recall@3 of ADA-002 is significantly better than SBERT ($p = 0.000$) for study questions, although not for quizzes ($p = 0.081$). Both ADA-002 and SBERT significantly outperform ConceptNet Numberbatch (both $p = 0.000$). The results are in line with [34], where contextual embeddings are used to represent entity descriptions in the entity typing task, and the results validate the superiority of ADA-002 in semantic representation over DistilBERT, a variation of the BERT model.

Table 3. Results of using embeddings derived from a single model. (**Bold**: max.).

	Quiz								**Study Question**							
	Recall@3				**MRR**				**Recall@3**				**MRR**			
	Bio.	**Soc.**	**Phy.**	**Mean**	**Bio.**	**Soc.**	**Phy.**	**Mean**	**Bio.**	**Soc.**	**Phy.**	**Mean**	**Bio.**	**Soc.**	**Phy.**	**Mean**
SBERT	**0.921**	0.977	0.957	0.951	0.724	0.922	0.814	0.820	0.864	0.980	0.908	0.917	0.721	0.907	0.758	0.795
ADA-002	0.909	**0.997**	**0.986**	**0.964**	**0.775**	**0.966**	**0.881**	**0.874**	**0.916**	**0.984**	**0.960**	**0.953**	**0.773**	**0.926**	**0.865**	**0.855**
CN-NB	0.830	0.951	0.918	0.899	0.665	0.897	0.755	0.773	0.742	0.856	0.827	0.808	0.610	0.760	0.691	0.687

Bio. = Biology, Soc. = Social Science, Phy. = Physics. CN-NB = ConceptNet Numberbatch.

4.2. Evaluation of Ensemble Models

Table 4 shows the results of the three strategies for creating ensembles of contextual embeddings from a pre-trained language model and static embeddings from ConceptNet Numberbatch. We divide our analysis of the results as follows: (i) evaluation within each group, i.e., combining either SBERT or ADA-002 with ConceptNet Numberbatch, (ii) evaluation of the projection-based ensembles, (iii) evaluation across groups, i.e., based on SBERT or ADA-002, and (iv) a summative analysis.

4.2.1. Combining Contextual Embeddings with ConceptNet Numberbatch

In the SBERT group, the late fusion ensemble, SBERT $+\!\!+$ CN-NB @0.5 consistently achieves the best results with respect to both Recall@3 and MRR in all subjects and both exercise types. Here, we take an unweighted average of the cosine similarity scores, i.e., $\alpha = 0.5$, so that the two models contribute equally. The McNemar test shows that the differences in Recall@3 between SBERT $+\!\!+$ CN-NB @0.5 and the single use of SBERT are statistically significant (quiz: $p = 0.029$, study question: $p = 0.000$). Additionally, SBERT $+\!\!+$ CN-NB @0.5 performs significantly better in Recall@3 than SBERT $\bigoplus$ CN-NB for study questions ($p = 0.000$), although not for quizzes ($p = 0.727$).

In the ADA-002 group, ADA-002 ⊕ CN-NB, i.e., a simple embedding vector concatenation in an early fusion fashion, exhibits the best results for all the mean performances, except slightly lower mean Recall@3 than using ADA-002 alone. A closer comparison between the results of the single use of ADA-002 and the best performer ADA-002 ⊕ CN-NB shows that the difference is very subtle and the advantages can almost be ignored. McNemar's test shows that the differences in Recall@3 between ADA-002 ⊕ CN-NB and using ADA-002 alone are not significant (quiz: $p = 1.000$, study question: $p = 0.203$). Moreover, ADA-002 ⊕ CN-NB outperforms ADA-002 + CN in all mean Recall@3s and MRRs.

4.2.2. Projecting Contextual Embeddings into ConceptNet Numberbatch Space

The results of learning a projection from contextual embeddings to ConceptNet Numberbatch embeddings are generally poor. The performance is significantly worse than any individual model of SBERT, ADA-002, or ConceptNet Numberbatch (all p-values for McNemar's test are 0.000 for the Recall@3s).

We find that the reason may lie in the fact that when projecting high-dimension vectors from a larger feature space (here, $Dim^{SBERT} = 768$, $Dim^{ADA-002} = 1536$) to the ConceptNet Numberbatch space ($Dim^{CN-NB} = 300$), we lose a large number of dimensions that may represent important contextual and semantic information. The reference experiment [41] that was able to achieve better results can be attributed to the fact that the embedding model used to project to ConceptNet Numberbatch is FastText, which has the same embedding dimension as ConceptNet Numberbatch. Therefore, the projection does not need to do any linear transformation, such as Principal Component Analysis, to reduce the vector to the same dimension or a nonlinear neural network learning for the mapping with the cost of losing valuable information. These things considered, the task in the reference experiment is a word/phrase level task, not a sentence level task as ours that needs to understand the context.

4.2.3. Comparing Ensembles with Different Contextual Embeddings

When we cross-analyze the SBERT and ADA-002 groups and compare the results of SBERT + CN-NB @0.5 (the best in the SBERT group) and the single use of ADA-002 (the best in the ADA-002 group), ADA-002 is shown to have better overall performance in the mean Recall@3 and MRR, except Recall@3 for quizzes. A further McNemar's test shows that the difference in Recall@3 for quizzes is not significant ($p = 1.000$).

4.2.4. Summative Analysis

In conclusion, using contextual embeddings from ADA-002 is shown to yield the overall best performance. The results not only validate the superiority of the ADA-002 model in semantic representation but also demonstrate that ADA-002 may have already embedded the concept and knowledge graph features that ConceptNet Numberbatch contains. Adding extra knowledge to SBERT can significantly boost the performance of the SBERT model through a linear interpolation of the similarity scores from each individual model. On the contrary, there is no significant gain when doing the same for ADA-002. The general performance of stacked embeddings of ADA-002 and ConceptNet Numberbatch is slightly better than using ADA-002 alone, but the advantage of this integration is very minor. The results align with previous work [34,40], where their empirical results also show that bringing in external knowledge through embeddings may not be useful or even harmful for certain tasks.

Furthermore, ADA-002 alone outperforms the ensemble of SBERT + CN-NB @0.5. In addition, learning the projection between embeddings may cause a loss of valuable information in the embeddings, especially when there is a reduction of dimensions as a result of the projection.

Table 4. Ensemble model outputs. (**Bold**: max. within each group).

	Quiz								Study Question							
	Recall@3				MRR				Recall@3				MRR			
	Bio.	**Soc.**	**Phy.**	**Mean**	**Bio.**	**Soc.**	**Phy.**	**Mean**	**Bio.**	**Soc.**	**Phy.**	**Mean**	**Bio.**	**Soc.**	**Phy.**	**Mean**
SBERT	0.920	0.976	0.957	0.951	0.724	0.922	0.814	0.820	0.864	0.980	0.908	0.917	0.721	0.907	0.758	0.795
SBERT ⇒ CN-NB	0.739	0.912	0.875	0.842	0.596	0.829	0.717	0.714	0.680	0.908	0.788	0.792	0.549	0.814	0.653	0.672
SBERT ⨁ CN-NB	0.921	0.977	0.964	0.954	0.729	0.920	0.798	0.816	0.866	0.980	0.911	0.919	0.720	0.911	0.757	0.796
SBERT ⨢ CN-NB @0.5	**0.928**	**0.995**	**0.971**	**0.965**	**0.770**	**0.951**	**0.843**	**0.855**	**0.902**	**0.986**	**0.944**	**0.944**	**0.743**	**0.925**	**0.804**	**0.824**
ADA-002	**0.909**	0.997	**0.986**	**0.964**	0.775	0.966	**0.881**	**0.874**	0.916	0.984	0.960	0.953	0.773	0.926	**0.865**	0.855
ADA-002 ⇒ CN-NB	0.470	0.684	0.607	0.587	0.398	0.552	0.543	0.498	0.466	0.565	0.559	0.530	0.399	0.485	0.472	0.452
ADA-002 ⨁ CN-NB	**0.909**	**1.000**	0.979	0.963	0.779	**0.971**	0.874	**0.874**	**0.918**	**0.989**	**0.969**	**0.959**	**0.784**	**0.937**	0.855	**0.859**
ADA-002 ⨢ CN-NB @0.5	0.894	**1.000**	0.968	0.954	**0.783**	0.964	0.840	0.862	0.869	0.978	0.931	0.926	0.729	0.907	0.794	0.810

Notation: Bio. = Biology, Soc. = Social Science, Phy. = Physics. CN-NB = ConceptNet Numberbatch. ⇒: learning a projection to, ⨁: concatenate embedding vectors, ⨢: interpolate the similarity scores. @0.5: $\alpha = 0.5$.

4.3. Linear Interpolation Analysis

Figure 5 shows the results of fusing the cosine similarities from the outputs of individual models through linear interpolation with different α values. When comparing the results from the two datasets, it can be noted that a very similar pattern of performance is achieved for quiz and study questions. The interpolants of fusing SBERT and ConceptNet Numberbatch reach peaks somewhere in the middle. The peak αs lie in 0.3, 0.5, 0.5, and 0.5, respectively, for the Recall@3 and MRR values of quizzes and study questions. This suggests that ConceptNet Numberbatch does bring value to SBERT, and the contributions are almost equivalent compared to SBERT. However, when it comes to ADA-002, the pattern is that the performance of the similarity fusion models perform better when αs increase. The αs that achieve the best performances are found at 1, 0.8, 0.9, and 0.9 for the Recall@3 and MRR values of quizzes and study questions. This implies that ConceptNet Numberbatch provides barely any value to ADA-002 in the similarity fusion model. When the fusion model reaches the highest Recall@3 for quizzes with $\alpha = 1$, it indicates that adding ConceptNet knowledge even harms the performance.

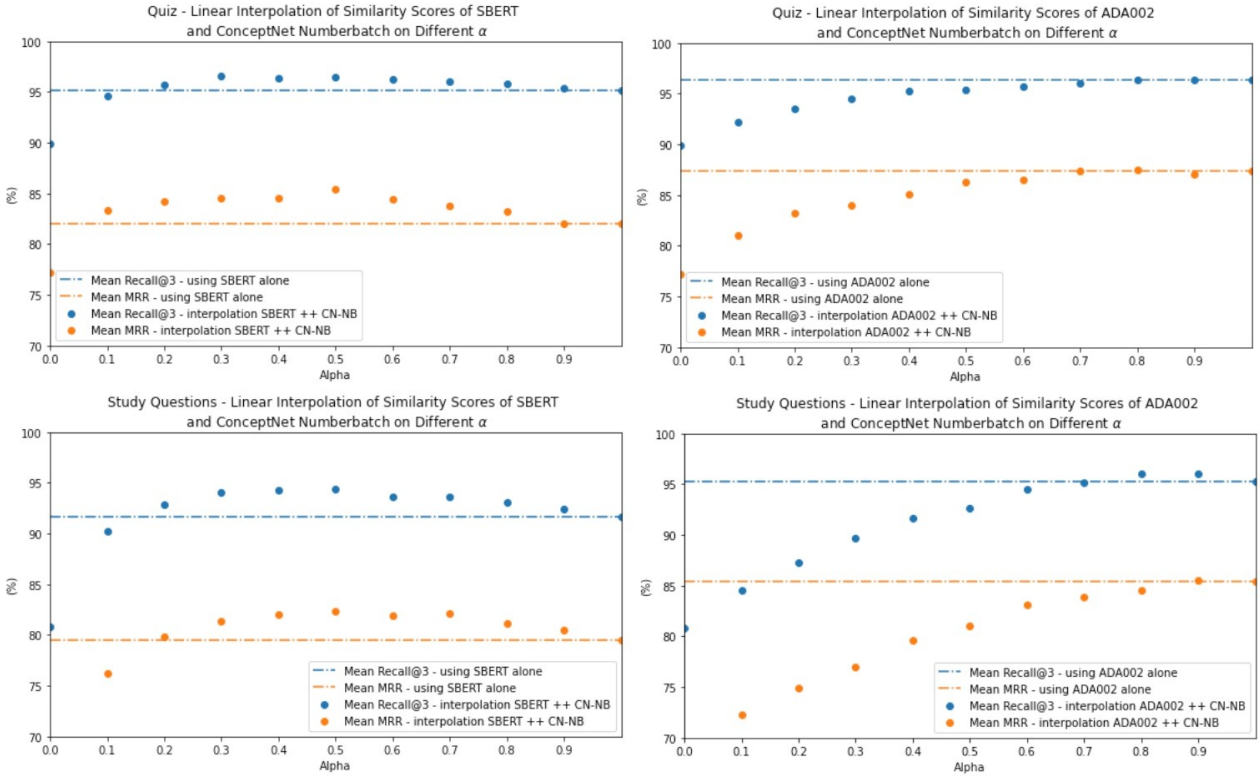


Figure 5. Linear interpolants between similarity scores when fusing SBERT | ADA002 and ConceptNet Numberbatch for different alphas.

4.4. Evaluation of ConceptNet with Domain-Specific Concept Filtering

ConceptNet collected knowledge from many sources, including Wiktionary, WordNet, etc., [27]. The concepts in ConceptNet vocabulary can be very trivial and can be distant from the real subject-wise educational domain. Therefore, when transforming ConceptNet's concept embeddings to representative concept-based sentence embeddings, some "concepts" in ConceptNet's vocabulary may not be educational domain concepts, possibly resulting in the sentence embeddings becoming diluted in their representation.

Based on this, we explore adding an extra layer to filter the identified concepts from the ConceptNet vocabulary. Such a concept filter can be built by training a domain-specific concept extraction model [59,60], or using some pre-trained "concept" extractor such as TagMe (entity linker) [61], DBpedia Spotlight (DBpedia hyperlinks), or TopicRank (keyphrase extraction) [62]. When it comes to terminology, there are some subtle differences for the real meaning for mentions, keywords, named entities, knowledge graph entities, concepts, educational concepts, or domain-specific educational concepts, which are the outputs from these "concept" extractors. Therefore, the objectives from different "concept" extractors can be different. Here, we instead use the gold-labeled concepts in the digital textbooks to construct a subject-wise domain concept dictionary. Figure 6 illustrates the idea of the ConceptNet model with domain-specific concept filtering. We posit that these filtered concepts are more relevant in the context of the input texts compared to the complete mapping of all concepts from the ConceptNet vocabulary.

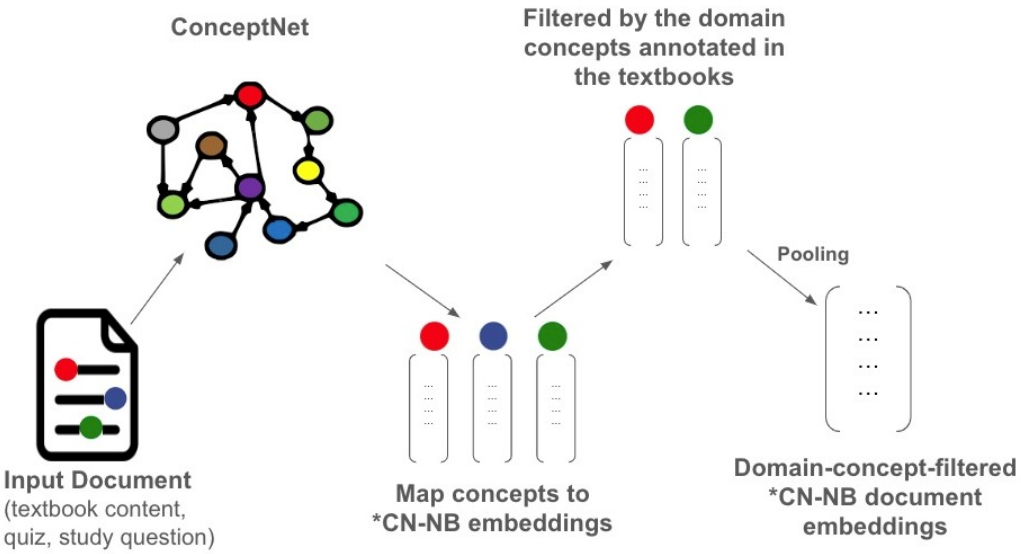


Figure 6. Illustration of using the ConceptNet Numberbatch model with domain-specific concept filtering. * CN-NB = ConceptNet Numberbatch.

Table 5 shows the results of using ConceptNet Numberbatch with domain-specific concept filtering. However, compared to using ConceptNet Numberbatch alone, domain-specific concept filtering does not improve the results. Figure 7 shows how textbook content is mapped to ConceptNet. It seems that almost all the words can be mapped, but "naturvetenskapliga" (English translation: natural scientific) as a compound word can only be found separately, although this is a common word in the Swedish language. Similarly, "Carl von Linné" is tagged word by word without having the famous whole name as a concept phrase. Some words ("stämband", "förtäting", and "förtunning") (English translation: "vocal cords", "thickening", and "thinning") are lacking inflectional forms. The results indicate that: (1) other non-core domain conceptual words can also provide useful information to the sentence embeddings, which explains why the original ConceptNet Numberbatch model performs better; (2) the training data of ConceptNet for the Swedish

language is not large enough to cover all words' inflectional forms, compound words, and phrases.

Table 5. Results of the ConceptNet model with domain-specific concept filtering.

	Quiz								Study Question							
	Recall@3				MRR				Recall@3				MRR			
	Bio.	**Soc.**	**Phy.**	**Mean**	**Bio.**	**Soc.**	**Phy.**	**Mean**	**Bio.**	**Soc.**	**Phy.**	**Mean**	**Bio.**	**Soc.**	**Phy.**	**Mean**
CN-NB	0.830	0.951	0.918	0.899	0.665	0.897	0.755	0.773	0.742	0.856	0.827	0.808	0.610	0.760	0.691	0.687
Filtered CN-NB	0.777	0.746	0.757	0.760	0.623	0.668	0.620	0.637	0.626	0.588	0.632	0.615	0.514	0.532	0.517	0.521

Bio. = Biology, Soc. = Social Science, Phy. = Physics. CN-NB = ConceptNet Numberbatch.

Example from Biology:

"Innan det naturvetenskapliga tänkandet och dess metoder blev allmänt accepterat av Carl von Linne och andra naturforskare, var det den bibliska skapelseberättelsen som bestämde synen på hur livet uppkom på jorden."

(English translation: Before the natural scientific thinking and its methods became widely accepted by Carl von Linnaeus and other natural scientists, it was the biblical creation story that determined the view of how life arose on earth.)

Example from Social Science:

"Tips! Månadens tema januari 2017: Sverige i säkerhetsrådet"

(English translation: Tip! Theme of the month January 2017: Sweden in the Security Council)

Example from Physics:

'När sångaren sjunger sätts stämbanden i svängning och luften börjar vibrera. Ljudvågorna fortplantas som förtätningar och förtunningar genom luften till ditt öra.'

(English translation: When the singer sings, the vocal cords are set in motion and the air begins to vibrate. The sound waves are propagated as thickenings and thinnings through the air to your ear.)

Figure 7. Examples of textbook content mapped to ConceptNet. Underlined: exists in ConceptNet. Red colored: part of word/phrase exists in ConceptNet.

4.5. Evaluation for Different Subjects and Exercise Types

Regarding subject matter, the general model performances across the three subjects follow the order: $SocialScience > Physics > Biology$. The models have better performance in Social Science than Natural Science in the STS tasks, which aligns with our previous study. The reason can be that natural science may involve more complex inferences and reasoning, with texts sometimes incorporating symbols, equations, or calculations, posing challenges for STS tasks.

In terms of exercise types, the mean Recall@3 and MRR values for quizzes consistently surpass those of study questions. The explanation can be that the answers to the quizzes provide more useful information for STS compared to study questions that contain no answers. Additionally, the best and worst mean performances generally stem from the same models for quizzes and study questions, respectively, as in the analysis in Sections 4.1–4.4.

5. Discussion

The results demonstrate that using powerful text representations in the form of embeddings derived from a large, pre-trained language model allows for the development of an STS-based content recommendation system that is capable of automatically and accurately linking exercises with textbook content. This approach is fully unsupervised, i.e., it does not require training or fine-tuning models based on task-specific labeled data, which is otherwise a major bottleneck in the development of AI, generally, but also in the educational domain and especially for smaller languages such as Swedish. Taking cost into account, the results, moreover, showed that combining two open-source models—i.e., SBERT and ConceptNet Numberbatch—in an ensemble can yield results that are competitive with those produced using OpenAI's proprietary embeddings model, ADA-002. In this setting, ensembles that combine the respective advantages of contextual embeddings and static

embeddings trained using a knowledge graph proved to be effective. However, the results showed that no significant benefit was obtained when having access to higher-dimensional and more powerful text embeddings, such as ADA-002.

We have presented an approach that can be used for resource-efficient development of an STS-based educational content recommendation system that was shown capable of accurately linking learning materials, here, in the form of textbook content and exercises. We argue that this forms a backbone component of an educational content recommendation system, which can be further developed by integrating student models and knowledge models, with the aim of providing personalized recommendations tailoring learning content to individual needs. Crucially, linking learning materials provides important value to learners, educators, and creators of digital learning materials. For learners, it can, for instance, help with self-regulated learning by recommending learning material—e.g., in the form of textbook content—that is relevant to an exercise and provide additional input to formative assessment activities. Especially when combined with student's performances on the exercises or assessment tests, for instance, if a student grapples with a specific concept elucidated in the textbook that is revealed through failing the corresponding related exercise or test question, the recommendation system can discern this challenge and proffer targeted learning content for remediation. This dynamic and adaptive learning path not only fosters deeper comprehension, but also bolsters engagement through real-time feedback and interactive assessment tools. Beyond individual student benefits, it can aid educators by pinpointing areas within a class or student group that require additional attention based on exercise or test performances. Educators can thereby, accordingly, refine instructional strategies, identify curricular gaps, and ensure that teaching methodologies align with students' evolving needs, which actively contribute to the pedagogical landscape. It can also be used by digital textbook creators in assessing the coverage of exercises with respect to textbook content and help to inform the design of exercises to align with chapter content. Thus, linking learning resources and building such AI-enhanced educational recommendation systems are foundational for intelligent textbook and technology-enhanced teaching and learning. However, to bridge the gap between technological expectations and user experiences, thorough testing, evaluation, and validation in real-world settings with educators and learners are imperative. Future work should expand testing to cover multiple subjects, languages, cross-cultural users, and diverse educational contexts to further enhance the pedagogical value of the proposed technical solution.

In this study, exercises were linked to textbook content at the level of textbook sections. This choice was primarily a result of the structure of the digital textbooks that were used, where there was a 1:1 mapping between exercises and textbook sections. While this arguably seems reasonable in terms of utility, one could consider recommending textbook content at different levels of granularity, e.g., in the form of chapters or sub-sections. The performance of linking exercises to textbook content at the chapter level would probably increase due to it likely being an easier task, while utility would most probably decrease. On the other hand, utility could potentially increase if recommendations were made on the sub-section or even paragraph level, depending on the nature of the exercise. In future work, we aim to evaluate our approach at different levels of abstraction, which will require manual annotation of the data or the involvement of human evaluators. Beyond different structural levels within textbook content, this technique can also be used to recommend and link various learning materials, such as curriculum goals, lecture slides, concept dictionaries, and external supporting resources. With the aid of an AI-enhanced content recommender, these learning materials can be harnessed effectively to cater to the requirements of both educators and students.

Our first attempt to evaluate Swedish knowledge graph-based ConceptNet Numberbatch model reveals that the model is trained under way too low a coverage of sources in the Swedish language, which results in an uncomplete Swedish vocabulary, leading to unstable and underperforming results. This may have an impact when using ConceptNet

Numberbatch alone or integrating with other models. The absence of a well-established pre-trained Swedish knowledge graph model leaves space for future research.

6. Conclusions

In this paper, we have evaluated the use of different pre-trained embeddings in an educational content recommendation system based on textual semantic similarity. The embeddings were derived from three types of models: (i) contextual embeddings from a pre-trained language model (SBERT), (ii) contextual embeddings from an LLM (ADA-002), and (iii) static embeddings trained using a concept-based knowledge graph (ConceptNet Numberbatch). In addition to evaluating the models individually, various ensembles were explored based on different strategies for combining pre-trained language models with a knowledge graph-based model, with the aim of leveraging the strengths of different models. The models were evaluated in the context of educational content recommendation, specifically in order to link two types of exercises (quizzes and study questions) to textbook content in an unsupervised fashion.

The contributions of this study are manifold, encompassing the empirical demonstration and evaluation of using pre-trained language models and knowledge graph-based model, individually and combined, when linking textbook content to exercises, the elucidation of methodologies for ensembling embeddings, and the derivation of valuable insights and practical guidance based on the results and analysis in the construction of educational recommendation systems. The experimental results revealed that leveraging the power of contextual embeddings from an LLM, i.e., ADA-002, produced the most accurate recommendations, in terms of Recall@3 and MRR, across three subjects (biology, social science, and physics) and for both quizzes and study questions. While the ensembles yielded improvements when combining contextual embeddings from SBERT and static embeddings trained using ConceptNet, there was no gain when contextual embeddings from an LLM, i.e., from ADA-002, were part of the ensemble. The performance of the best-performing model was high for both types of exercises, resulting in a mean Recall@3 of 0.96 for quizzes and 0.95 for study questions, which means that, for 95–96% of exercises, the correct textbook section was identified in the top three recommendations. It was also observed that the models have somewhat better performance in social science than in natural science subjects. The strong performance demonstrates the feasibility of using STS based on embeddings from pre-trained language models for an educational content recommendation, laying the foundation for developing a comprehensive personalized adaptive content recommendation system.

Author Contributions: Conceptualization, X.L. and A.H.; methodology, X.L., A.H., M.D. and J.N.; software, X.L.; validation, X.L., A.H., M.D., J.N. and Y.W.; formal analysis, X.L., A.H., M.D., J.N. and Y.W.; resources, X.L. and A.H.; writing—X.L. and A.H.; writing—review and editing, X.L., A.H., M.D., J.N. and Y.W.; visualization, X.L.; supervision, A.H., M.D. and J.N.; project administration, A.H.; funding acquisition, J.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are unavailable due to privacy or ethical restrictions since the digital textbook data are from a commercial publisher. The source code can be found at the following URL: https://github.com/lixiu911/future_internet_journal, accessed on 30 November 2023.

Acknowledgments: Special thanks to my main supervisor, Aron Henriksson, for the support during the project.

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

References

1. Thaker, K.; Zhang, L.; He, D.; Brusilovsky, P. Recommending Remedial Readings Using Student Knowledge State. In Proceedings of the International Conference on Educational Data Mining (EDM), Online, 10–13 July 2020; International Educational Data Mining Society: Worcester, MA, USA, 2020.

2. Okubo, F.; Shiino, T.; Minematsu, T.; Taniguchi, Y.; Shimada, A. Adaptive Learning Support System Based on Automatic Recommendation of Personalized Review Materials. *IEEE Trans. Learn. Technol.* **2022**, *16*, 92–105. [CrossRef]
3. 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 Second International Workshop on Intelligent Textbooks 2020, Ifrane, Morocco, 6 July 2020.
4. Rahdari, B.; Brusilovsky, P.; Thaker, K.; Barria-Pineda, J. Knowledge-driven wikipedia article recommendation for electronic textbooks. In Proceedings of the European Conference on Technology Enhanced Learning, Heidelberg, Germany, 14–18 September 2020; pp. 363–368.
5. Barria-Pineda, J.; Narayanan, A.B.L.; Brusilovsky, P. *Augmenting Digital Textbooks with Reusable Smart Learning Content: Solutions and Challenges*; Technical Report; EasyChair: Manchester, UK, 2022.
6. Herlinda, R. The use of textbook in teaching and learning process. In Proceedings of the 6th TEFLIN International Conference, Surakarta, Indonesia, 7–9 October 2014.
7. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
8. OpenAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.
9. Liétard, B.; Abdou, M.; Søgaard, A. Do Language Models Know the Way to Rome? *arXiv* **2021**, arXiv:2109.07971.
10. Hadi, M.U.; Qureshi, R.; Shah, A.; Irfan, M.; Zafar, A.; Shaikh, M.; Akhtar, N.; Wu, J.; Mirjalili, S. A survey on large language models: Applications, challenges, limitations, and practical usage. *TechRxiv* **2023**. [CrossRef]
11. Zhao, X.; Lu, J.; Deng, C.; Zheng, C.; Wang, J.; Chowdhury, T.; Yun, L.; Cui, H.; Xuchao, Z.; Zhao, T.; et al. Domain Specialization as the Key to Make Large Language Models Disruptive: A Comprehensive Survey. *arXiv* **2023**, arXiv:2305.18703.
12. Rajabi, E.; Etminani, K. Knowledge-graph-based explainable AI: A systematic review. *J. Inf. Sci.* **2022**. [CrossRef]
13. Zhang, F.; Yuan, N.J.; Lian, D.; Xie, X.; Ma, W.Y. Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 353–362.
14. 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]
15. Xiong, C.; Power, R.; Callan, J. Explicit semantic ranking for academic search via knowledge graph embedding. In Proceedings of the 6th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1271–1279.
16. Li, X.; Henriksson, A.; Nouri, J.; Duneld, M.; Wu, Y. Linking Swedish Learning Materials to Exercises through an AI-Enhanced der System. In Proceedings of the International Conference in Methodologies and Intelligent Systems for Techhnology Enhanced Learning, Guimaraes, Portugal, 12–14 July 2023; pp. 96–107.
17. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21– 26 June 2014; pp. 1188–1196.
18. Niu, Y.; Lin, R.; Xue, H. Research on Learning Resource Recommendation Based on Knowledge Graph and Collaborative Filtering. *Appl. Sci.* **2023**, *13*, 10933. [CrossRef]
19. Sterling, J.A.; Montemore, M.M. Combining Citation Network Information and Text Similarity for Research Article Recommender Systems. *IEEE Access* **2021**, *10*, 16–23. [CrossRef]
20. Ostendorff, M. Contextual document similarity for content-based literature recommender systems. *arXiv* **2020**, arXiv:2008.00202.
21. Wan, S.; Niu, Z. A hybrid e-learning recommendation approach based on learners' influence propagation. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 827–840. [CrossRef]
22. Rahman, M.M.; Abdullah, N.A. A personalized group-based recommendation approach for Web search in E-learning. *IEEE Access* **2018**, *6*, 34166–34178. [CrossRef]
23. Vrandečić, D.; Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM* **2014**, *57*, 78–85. [CrossRef]
24. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
25. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 10–12 June 2008; pp. 1247–1250.
26. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In Proceedings of the International Semantic Web Conference, Busan, Republic of Korea, 11–15 November 2007; pp. 722–735.
27. Speer, R.; Chin, J.; Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
28. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
29. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *arXiv* **2017**, arXiv:1706.02216.
30. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June 2018; pp. 593–607.

31. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
32. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
33. Wang, Z.; Li, J.; Liu, Z.; Tang, J. Text-enhanced representation learning for knowledge graph. In Proceedings of the International Joint Conference on Artificial Intelligent (IJCAI), New York, NY, USA, 9–15 July 2016; pp. 4–17.
34. Zhao, Q.; Lei, Y.; Wang, Q.; Kang, Z.; Liu, J. Enhancing text representations separately with entity descriptions. *Neurocomputing* **2023**, *552*, 126511. [CrossRef]
35. Yu, D.; Zhu, C.; Yang, Y.; Zeng, M. Jaket: Joint pre-training of knowledge graph and language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; Volume 36, pp. 11630–11638.
36. Yamada, I.; Asai, A.; Shindo, H.; Takeda, H.; Matsumoto, Y. Luke: Deep contextualized entity representations with entity-aware self-attention. *arXiv* **2020**, arXiv:2010.01057.
37. El Boukkouri, H.; Ferret, O.; Lavergne, T.; Zweigenbaum, P. Embedding strategies for specialized domains: Application to clinical entity recognition. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Florence, Italy, 28 July–2 August 2019; pp. 295–301.
38. Wang, B.; Shen, T.; Long, G.; Zhou, T.; Wang, Y.; Chang, Y. Structure-augmented text representation learning for efficient knowledge graph completion. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–13 April 2021; pp. 1737–1748.
39. Muromägi, A.; Sirts, K.; Laur, S. Linear ensembles of word embedding models. *arXiv* **2017**, arXiv:1704.01419.
40. Gammelgaard, M.L.; Christiansen, J.G.; Søgaard, A. Large language models converge toward human-like concept organization. *arXiv* **2023**, arXiv:2308.15047.
41. Goossens, S. A Guide to Building Document Embeddings. Available online: https://radix.ai/blog/2021/3/a-guide-to-building-document-embeddings-part-1/ (accessed on 28 November 2023).
42. Ganaie, M.A.; Hu, M.; Malik, A.; Tanveer, M.; Suganthan, P. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [CrossRef]
43. Faruqui, M.; Dodge, J.; Jauhar, S.K.; Dyer, C.; Hovy, E.; Smith, N.A. Retrofitting word vectors to semantic lexicons. *arXiv* **2014**, arXiv:1411.4166.
44. Fang, L.; Luo, Y.; Feng, K.; Zhao, K.; Hu, A. Knowledge-enhanced ensemble learning for word embeddings. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 427–437.
45. Fang, L.; Luo, Y.; Feng, K.; Zhao, K.; Hu, A. A Knowledge-Enriched Ensemble Method for Word Embedding and Multi-Sense Embedding. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 5534–5549. [CrossRef]
46. Zhong, Q.; Ding, L.; Liu, J.; Du, B.; Jin, H.; Tao, D. Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 10098–10111. [CrossRef]
47. Ri, R.; Yamada, I.; Tsuruoka, Y. mLUKE: The power of entity representations in multilingual pretrained language models. *arXiv* **2021**, arXiv:2110.08151.
48. Peters, M.E.; Neumann, M.; Logan IV, R.L.; Schwartz, R.; Joshi, V.; Singh, S.; Smith, N.A. Knowledge enhanced contextual word representations. *arXiv* **2019**, arXiv:1909.04164.
49. Malmsten, M.; Börjeson, L.; Haffenden, C. Playing with Words at the National Library of Sweden–Making a Swedish BERT. *arXiv* **2020**, arXiv:2007.01658.
50. Rekathati, F. The KBLab Blog: Introducing a Swedish Sentence Transformer. 2022. Available online: https://kb-labb.github.io/posts/2021-08-23-a-swedish-sentencetransformer (accessed on 28 November 2023).
51. Isbister, T.; Sahlgren, M. Why not simply translate? A first Swedish evaluation benchmark for semantic similarity. *arXiv* **2020**, arXiv:2009.03116.
52. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
53. Neelakantan, A.; Xu, T.; Puri, R.; Radford, A.; Han, J.M.; Tworek, J.; Yuan, Q.; Tezak, N.; Kim, J.W.; Hallacy, C.; et al. Text and code embeddings by contrastive pre-training. *arXiv* **2022**, arXiv:2201.10005.
54. Greene, R.; Sanders, T.; Weng, L.; Neelakantan, A. New and Improved Embedding Model. OpenAI Blog. Available online: https://openai.com/blog/new-and-improved-embedding-model (accessed on 28 November 2023).
55. Ekgren, A.; Gyllensten, A.C.; Gogoulou, E.; Heiman, A.; Verlinden, S.; Öhman, J.; Carlsson, F.; Sahlgren, M. Lessons Learned from GPT-SW3: Building the First Large-Scale Generative Language Model for Swedish. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 20–25 June 2022; pp. 3509–3518.
56. Mihalcea, R.; Corley, C.; Strapparava, C. Corpus-based and knowledge-based measures of text semantic similarity. In Proceedings of the AAAI, Boston, MA, USA, 16–20 July 2006; Volume 6, pp. 775–780.
57. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
58. Su, Y.; Kabala, Z.J. Public Perception of ChatGPT and Transfer Learning for Tweets Sentiment Analysis Using Wolfram Mathematica. *Data* **2023**, *8*, 180. [CrossRef]
59. Meng, R.; Zhao, S.; Han, S.; He, D.; Brusilovsky, P.; Chi, Y. Deep keyphrase generation. *arXiv* **2017**, arXiv:1704.06879.

60. Li, X.; Nouri, J.; Henriksson, A.; Duneld, M.; Wu, Y. Automatic Educational Concept Extraction Using NLP. In Proceedings of the International Conference in Methodologies and Intelligent Systems for Techhnology Enhanced Learning, L'Aquila, Italy, 12–14 July 2022; pp. 133–138.
61. Ferragina, P.; Scaiella, U. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In Proceedings of the 19th ACM International Conference on INFORMATION and Knowledge Management, Toronto, ON, Canada, 26–30 October 2010; pp. 1625–1628.
62. Bougouin, A.; Boudin, F.; Daille, B. Topicrank: Graph-based topic ranking for keyphrase extraction. In Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), Nagoya, Japan, 14–19 October 2013; pp. 543–551.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.