Topic Analysis of Textbooks Using AI

Coby B. Simmons
cobysimmons01@gmail.com

Utrecht University
The University of Sydney

February 2024

Abstract

Advances in the field of artificial intelligence (AI) and natural language processing (NLP) have opened new horizons for personalised learning. It has been shown that online learning platforms are tremendously beneficial in promoting student engagement. Given the widespread use of textbooks in the learning process and the limited technological advancements that have been made with regard to their use, it is worth exploring how textbooks can be leveraged to enhance the learner's experience in the growing trend towards online learning. This research focuses on the applications, of AI in identifying and categorising topics within educational content. Specifically, we aim to construct comprehensive models that accurately reflect the depth and breadth of a particular domain using the knowledge embedded in textbooks. These domain-specific and fine-grained topic models will be used to generate a dataset for the training of state-of-the-art deep learning models to identify the topics present in new pieces of text. Our motivation is that a dense and comprehensive domain model based on textbooks can serve as a sophisticated framework for educational applications, enabling them to provide enhanced learning experiences via downstream tasks such as learning object classification, automatic index creation, and content personalisation. Thus, we propose an architecture that moves away from classic, unsupervised topic modelling techniques and attempts to implement a fine-grained, semi-supervised approach.

Keywords: textbooks, semantic linking, natural language processing, deep learning, semi-supervised learning, topic classification

1 Introduction

The advent of online learning has brought about a groundbreaking transformation in education, offering unparalleled opportunities to learners worldwide. It has been shown that online learning platforms are tremendously beneficial in promoting student engagement [10]. Given the widespread use of textbooks in the learning process [32], it is worth exploring how they can be leveraged to enhance the learner's experience in the growing trend towards online learning.

Work to bring textbooks into the digital age has already been undertaken. Specifically, there exists online learning platforms that incorporate improved search and navigation functionalities [22, 50, 14, 13] and interactive content [25]. However, in practice, these methods are typically heuristic and do not necessarily take advantage of AI. Additionally, for most textbooks, the digital and printed versions are identical. This may be because the AI models underlying these e-learning approaches do not exhibit the high level of performance required for acceptance in the rigorous domain of education.

For AI to perform effectively on education-related tasks, it is necessary to construct a domain-specific, fine-grained topic model. "Domain-specific", in this context, emphasises that concepts or knowledge are unique in their definition for a particular domain [48]. Consider the concept of "mean" as an example. In the statistics domain, the mean is typically related to probability distributions, statistical inference, the Central Limit Theorem, and so on. In finance, however, the concept is more closely associated with the notions of risk or time value of money. Given that textbooks (and other learning objects) are generally limited to their respective domains, this research investigates whether domain-specificity can help improve the accuracy and efficiency of AI models by imparting a better understanding of relevance.

Working in conjunction with domain-specificity is the notion of fine granularity. In the context of topic models, fine granularity refers to the detailed and specific division of topics, where the model identifies and differentiates between closely related subtopics rather than grouping them under broad, general categories. This approach allows for a more precise and nuanced understanding of the content, enabling the identification of subtle distinctions and variations within the data.

A domain-specific, fine-grained topic model may not be essential for broader applications of classification. For example, when classifying topics in news articles, the topics tend to be broader and less nuanced (e.g. politics, sports,

business), allowing for easy classification without the need for deep, specialised analysis [35].

Conversely, academic texts often delve into complex and highly specialised topics, where differences are subtler and lexical overlap is significant [5, 60]. This level of detail requires a fine-grained approach to accurately classify and organise content according to its precise academic domain, facilitating better inference outcomes. Thus, our research is based on the premise that domain-specific, fine-grained topic models are essential in capturing the depth and breadth of subjects in textbooks.

Our motivation is that a dense and comprehensive domain model based on textbooks can serve as a sophisticated framework for educational applications, enabling them to provide enhanced learning experiences via downstream tasks such as learning object classification, automatic index creation, and content personalisation. This application of AI in education has the potential to revolutionise the way we teach and learn, making education more efficient, effective, and accessible for all.

Given this, our overarching research question is

RQ0: What is the efficacy of training fine-grained, domain-specific topic models from textbooks using machine learning methods?

We explore the optimal configuration for such an approach with three sub-research questions, the first of which is

RQ1: Which topic modelling techniques perform best in linking content to construct a topic hierarchy?

This content linkage process leads to the creation of a fully integrated textbook that covers the breadth and depth of a particular domain. This "joint hyperspace of cross-linked textbooks" is introduced to address issues such as subjectivity, limited scope and inadequate detail that may arise in individual textbooks, thus resulting in "a more complete and objective model" [4].

This comprehensive and unified data collection can be normalised and then used to train deep learning models to classify topics within the domain of choice. We consider the conditions that lead to optimal performance in this stage of the process with two more sub-research questions:

RQ2: How does the performance of deep learning models vary when additional semantic information (i.e. concept annotations) is included in the data?

RQ3: How does the performance of deep learning models vary across datasets with different levels of quality?

This research shows that topic models extracted from the hierarchy of textbooks perform better than the baseline at classifying new textbook sections into topics, particularly when additional semantic information is included in the data. Our proposed method for automatic content linkage demonstrated promising results, achieving a performance level that, while not yet matching the precision of experts' manual efforts, shows significant potential. It's important to note that there was a decrease in topic classification accuracy when utilising the automatically generated topic model, suggesting areas for further refinement.

The rest of this paper is organised as follows: Section 2 provides an overview of the existing research in this space, with a particular focus on the methods that we will apply. In Section 3, our methodology is further discussed with a detailed proposal for an architecture that can be used to answer our research questions. Our results are explored and evaluated in Section 4 and they are discussed in Section 5. Finally, Section 6 summarises the paper and indicates some areas for future research.

2 Background

2.1 Modelling of Textbooks

Previous works have explored the transformation of PDF textbooks into interactive and intelligent online learning materials [2, 3, 4, 5]. Figure 1 broadly summarises these works as a single automatic workflow that has been developed to extract knowledge models for PDF textbooks.

First, a comprehensive set of rules is defined to capture the typical conventions in the formatting, structure, and organisation of textbooks. Key elements such as the structure of the textbook (including chapters and subchapters), content (consisting of words, lines, text fragments, pages and sections) and domain-specific terms are extracted.

The workflow then uses the domain terms to connect to DBpedia, a large-scale knowledge base extracted from Wikipedia [36]. The linkage of glossary terms to DBpedia facilitates the enrichment of these terms with semantic information such as concept definitions and categories (see Section 3 for more detail), thus improving the quality of the domain terms, broadening their contextual understanding, and connecting the textbooks to the Linked Open Data Cloud.

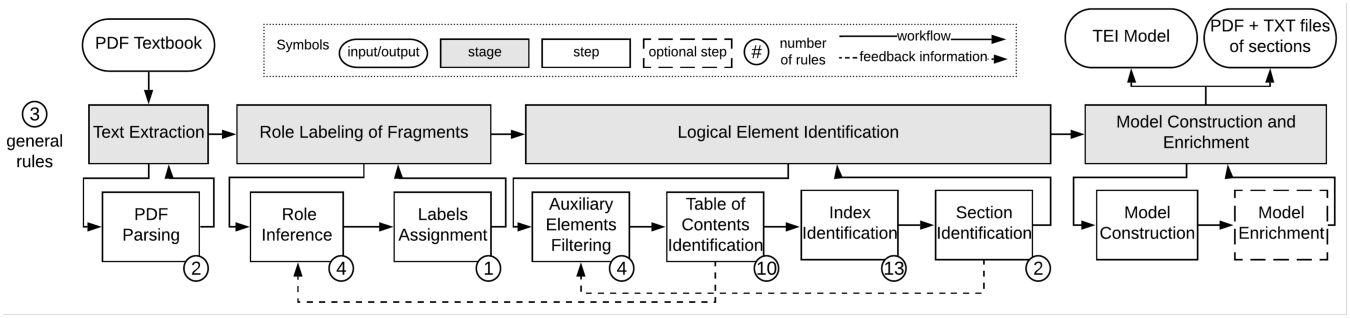
Finally, the extracted knowledge is serialised according to the Text Encoding Initiative (TEI) Guidelines[1] and the resulting files are used as the input for this research (see Section 3.1.1).

The TEI Guidelines establish a unique XML-based format that predominantly focuses on the semantic aspects of the textbook, unlike other open file formats that focus more on presentation (e.g. HTML, PDF, etc.). The framework encompasses around 500 distinct textual elements (e.g. words or glyphs) and concepts (e.g. names or definitions).

Furthermore, "back-of-the-book indices" are explored for the benefits that they provide in information retrieval and

[1] https://www.tei-c.org/release/doc/tei-p5-doc/en/html/index.html

Figure 1: Textbook knowledge model construction workflow from Alpizar-Chacon and Sosnovsky [4]



knowledge organisation since they provide a collection of important domain terms curated by experts. The research presents a method for enriching the knowledge model of textbooks, by classifying each index term according to their relevance to a particular domain. Each term can be classified as

- **core-domain:** index terms representing the most important and frequent concepts in the domain;
- **in-domain:** additional index terms in the domain;
- **related-domain:** index terms tangential to the domain;
- **out-of-domain:** index terms not related to the domain (e.g. from examples).

This annotation of concepts found in the index can be exploited to aid in the modelling of textbook topics. By incorporating these concepts, which succinctly capture the essence of each section, the model gains access to a concise summary that highlights the main ideas. This additional layer of information is anticipated to improve the model's ability to accurately identify and weigh the significance of crucial terms within the text [44, 6].

While the above workflow describes the construction of a textbook knowledge model, it does not specify how learners might be able to access this knowledge effectively and engagingly. Thus, a proposal is made for 'Intextbooks', an intelligent and personalised online learning experience [7]. In this proposal, 'semantic markup' is closely integrated with a presentational representation of the PDF textbook to provide a platform where a learner can not only view a given textbook, but can also jump to different sections, search for occurrences of a word, and access enriched semantic information that is external to the textbook itself. Aside from providing a user interface for the domain and content model, a user might be able to have conversations with other users or annotate the textbook's content with highlights, bookmarks, and notes. Thus, Intextbooks results in an e-learning platform that enriches the simple and somewhat featureless PDF textbook.

2.2 Topic Modelling & Semantic Similarity

In his foundational treatise on logic and argumentation, *Topics*, Aristotle develops a system of 'places' – mental locations or categories – where different types of arguments could be found and utilised. Thus, the term 'topic' comes from the Greek *tópos* (τόπος), meaning 'place' or 'locus' [23].

Topics can be formed and detected through a variety of processes, but perhaps the most significant source is through inquiry and discourse. They are often shaped by the way we describe and talk about phenomena. Textbooks are the ultimate resource for these processes [45]. Thus, they can serve as an effective input into the modelling of topics [4].

Mathematical concepts like vectors and probability can be used to model the abstract structure of topics, just as they can be used to model other spaces (both physical or abstract) [16]. These "vector space or semantic space models of meaning" underlie nearly all modern methods in semantics that involve corpus analysis [12]. Thus, a word's meaning is represented by a vector that records frequency and co-occurrence patterns within text corpora.

Semantic similarity refers to the degree to which two pieces of text are alike in meaning [40]. Within the aforementioned frameworks, the similarity between two or more words can be accurately measured by a geometric comparison of the vectors that represent them. The cosine similarity metric is used in this research since it has been shown previously to be a very effective measure of semantic similarity [15].

Given two n-dimensional semantic vectors, $\mathbf{a}$ and $\mathbf{b}$, the cosine similarity S_C is the dot product of the vectors, divided by the product of their lengths. Additionally, cosine similarity can also be thought of as the angle θ between the two vectors. Both interpretations are shown in Equation 1.

$$S_C(\mathbf{a}, \mathbf{b}) := \cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} \tag{1}$$

Thus, the directions of the two vectors serve as a proxy for their semantic content. A cosine similarity of 0 indicates

orthogonal vectors, representing no semantic similarity. A cosine similarity of 1 or -1 indicates parallel vectors, representing semantic content is perfectly similar or perfectly opposite, respectively.

Cosine similarity is unaffected by the magnitude of vectors, a particularly useful property in text analysis where the length of documents can vary greatly.

In recent years, various techniques for unsupervised topic discovery have been introduced. The most relevant of these techniques are outlined in the remainder of this section.

2.2.1 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a statistical measure used to evaluate the importance of a word in a document, in relation to the document's corpus. The Term Frequency (TF) component measures the frequency of a word in a specific document, essentially representing the importance of the word in understanding the document's content. The Inverse Document Frequency (IDF) component is the logarithmically-scaled, inverse of the frequency of documents in the corpus that contain a particular term, penalising terms that appear in many documents, as they are less valuable in distinguishing between documents [51]. Thus, for a term t in a document d in a corpus of size N, the TF-IDF is

$$tf_{t,d} \times \log \frac{N}{df_t} \tag{2}$$

where $tf_{t,d}$ is the frequency of t in d and df_t is the number of documents containing t.

2.2.2 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is another foundational technique in the development of topic models [18]. At its core, LSI aims to identify patterns in the relationship between terms, documents, and concepts that simple term matching cannot pick up. For example, TF-IDF often fails to account for nuances like synonyms or context.

These latent semantic structures are uncovered by applying Singular Value Decomposition (SVD) to the original term-document matrix. SVD decomposes this matrix into three smaller matrices, representing the relationships between documents, terms, and the concepts they imply. These matrices are then recombined in a way that retains most of the significant information while discarding noise and less important details. The essential patterns captured in this reduced matrix reveal how terms and documents are related in terms of underlying topics or concepts. By focusing on these patterns rather than just exact term matches, LSI more effectively aligns queries with relevant documents, improving the accuracy of information retrieval.

2.2.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA), an extension of LSI, was introduced by Blei, Ng, and Jordan [11] as a generative probabilistic model for topic modelling. In LDA, each topic is defined by the likelihood of each word appearing in that topic (e.g. the topic 'physics' might assign higher probabilities to words like 'quantum', 'energy', or 'particle'). In turn, documents are modelled as as mixture of topics (e.g. a document in a collection of academic texts might be 30% about physics, 50% about mathematics, and 20% about computer science).

In Guerra, Sosnovsky, and Brusilovsky [29], LDA was used to link documents from different textbooks. It was found to be more effective than a baseline method that uses Apache Lucene based on the TF-IDF model.

However, a significant issue with LDA is the need to predefine the number of topics. This parameter can vary greatly for different types of datasets and can also significantly influence the model's performance and the coherence of the topics generated. Furthermore, LDA may produce poor-quality topics when documents have overlapping or tangential topics [1].

2.2.4 Embedding Models

In recent years, embedding models have revolutionised the field of topic modelling, moving beyond traditional methods that treat words and topics as discrete, atomic units. These models create dense vector representations of words and documents, capturing semantic relationships in a continuous vector space, and thus providing a more sophisticated and contextually aware approach to understanding language and text data.

Word2Vec is a pivotal example of this approach [39]. It generates vector representations for words in such a way that the spatial relationships between vectors capture semantic relationships between words. For instance, in this vector space, words with similar meanings, like 'apple' and 'banana,' are positioned closer together than unrelated words like 'elephant.' Word2Vec also excels at capturing analogies; for example, the relationship between 'king' and 'queen' is analogous to that between 'man' and 'woman' in its vector space.

Extending this concept to documents, Doc2Vec allows entire documents to be represented as vectors [34]. This method enables the comparison of documents in a multi-dimensional vector space that considers the overall context and semantic content of the documents, not just isolated word usage. Doc2Vec outperforms LDA in textbook content linkage tasks [53].

Another significant advancement in this field is Top2Vec [8]. Top2Vec builds upon the principles of Word2Vec and Doc2Vec by creating joint embeddings of words and documents. It automatically detects topic vectors in the joint word and document embedding space. Each detected topic is represented as a dense vector in this space, and docu-

ments and words closest to this vector are most relevant to the topic. This approach allows for a more intuitive and semantically rich way of understanding and categorising documents and topics. It effectively captures the nuances of language and thematic structures within large text corpora.

2.3 Advanced Language Models

In recent years, more powerful and advanced language models have become available.

2.3.1 Bidirectional Encoder Representations from Transformers

In Devlin et al. [20], Bidirectional Encoder Representations from Transformers (BERT) are introduced as a "language representation model" that operates bidirectionally, i.e. it can understand the context of words in a sentence from both the left and right directions. This represents a significant leap forward from previous models that perceive text unidirectionally.

The lifecycle of preparing a BERT model for an NLP task is divided into the pre-training and fine-tuning stages. These processes are displayed graphically in Figure 2.

BERT is unique in its approach to pre-training, which involves two unsupervised tasks. It uses masked language modelling, where some percentage (usually 15%) of input tokens are replaced by random masking and the model is trained to predict these masked tokens. Additionally, BERT utilises a next-sentence prediction (NSP) task to determine if a sentence follows the previous one in the original document. This method enables the model to understand sentence relationships and context, which is crucial for many downstream tasks. Pre-training typically uses a large corpus of texts (e.g. Wikipedia). It can take hours to days and requires significant computing power. Thus, various open-source pre-trained models are available for download [58].

In the fine-tuning phase, the pre-trained BERT model is adjusted to a more specific NLP task (e.g. sentiment analysis, question answering, or document classification). This involves adjusting the model's weights slightly to tailor it to the nuances of the task at hand, leveraging the general understanding of language that the model gained during pre-training. Fine-tuning is typically performed on a smaller, narrower, labelled dataset.

BERT has been shown to achieve state-of-the-art performance when the language embeddings that it generates are used as input features for downstream NLP tasks.

In the education domain, the rich data potential of textbooks has been harnessed by using BERT to develop a tool for automatic keyword extraction, thus providing a cost-effective and efficient alternative to traditional manual data annotation methods [44]. Given the existing workflow for textbook knowledge model extraction described in Section 2.1, this approach leverages textbook indexes to extract domain-specific knowledge that can be used to label datasets, which in turn can be used for downstream supervised learning tasks.

It is important to note that BERT is extremely computationally intensive. To mitigate this, research was undertaken to develop a method to pre-train a distilled version of BERT (DistilBERT) [47]. It was shown that DistilBERT can result in a BERT model that is 40% smaller and 60% faster while retaining 97% of the language understanding capabilities of the original model.

2.3.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of deep neural network that functions as a dynamic system, possessing a distinct internal state at every stage of the classification process [52]. This unique characteristic stems from self-feedback loops within the network, which facilitate the transfer of information from previous occurrences to the current processing stages. Consequently, RNNs develop a capacity to remember sequences in the data.

However, RNNs can face the "vanishing gradient problem", whereby the gradient of the loss function becomes too small for effective learning, making it difficult for the network to learn long-range dependencies [9].

Additionally, the sequential nature of RNNs means they cannot be easily parallelised, leading to longer training times compared to other architectures. While RNNs can theoretically remember long sequences, in practice, they often struggle with dependencies that are separated by long intervals in the data.

Long Short-Term Memory (LSTM) networks were introduced to overcome the limitations of traditional RNNs, particularly the vanishing gradient problem [30]. This is achieved through the introduction of a structure called a memory cell, capable of maintaining information in memory for long periods. The key to this capability is the use of gates that regulate the flow of information into and out of the cell, thereby addressing the vanishing gradient problem and preserving long-term dependencies in sequential data more effectively than traditional RNNs.

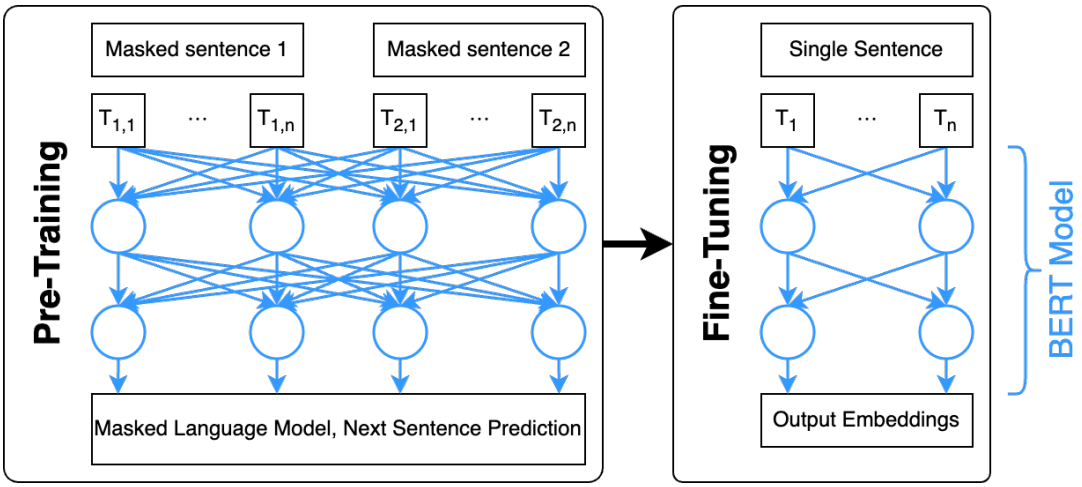
2.4 Linking Multiple Textbooks

Guerra, Sosnovsky, and Brusilovsky [29] describes two methods for handling the hierarchical structure of the textbook, which can be applied to our vector representations of documents:

a) **Topic Aggregation:** Only compute the topic vectors for the sections at the lowest level in the hierarchy, and for higher levels, aggregate topic vectors by taking a weighted average of sub-topic vectors.

b) **Re-Indexing:** Compute the topic vector for every section by considering a section's 'document' to be the content of the section and all its sub-sections.

Figure 2: Diagram of the BERT pre-training and fine-tuning stages



The Topic Aggregation method is much more computationally efficient since vectors do not need to be recomputed regularly. Thus, it is the approach taken for this research.

2.5 Topic Classification

Topic classification involves categorising texts or documents into predefined topics or classes. Typically, this classification has been performed across broad and distinctly separate areas (e.g. such as technology, sports, and science) with considerable success [35]. Such categorisation leverages the clear conceptual and lexical differences between categories, facilitating the development of models that can accurately differentiate based on distinct features and terminologies inherent to each domain. As described in Section 2.2.3, LDA is primed to perform well under these conditions and there is evidence of this [33, 37].

However, challenges emerge when topic classification is applied within a single, homogeneous domain, where the distinctions between subtopics are subtler and the lexical overlap is significant. Fine-grained classification seeks to distinguish between smaller, more specific categories that fall under a broader parent domain [60]. These specific categories are generally defined by domain experts using intricate criteria, which generally focus on subtle differences. Synthesising these definitions is a time-consuming task. However, we propose that the hierarchy of textbooks can effectively be translated into a model of the topics in a domain. Through the content linkage process defined in Section 3.1.2, a comprehensive and unified data collection for a particular domain is generated, thus removing the need for resource-intensive, fine-grained human annotations.

2.6 Summary

Inspired and informed by the related works summarised in Table 1, we aim to contribute to the field of AI in education through our proposal of a new architecture. This architecture builds upon principles used in fine-grained image classification to generate a labelled dataset with minimal human intervention that is then used as the input for a domain-specific topic classification model.

3 Methodology

This section proposes our methodology for the extraction of fine-grained, domain-specific topic models from textbooks to answer **RQ0**. Our motivation for this method is that it is a quick and inexpensive way to develop a dataset that provides a rich and fine-grained representation of a particular subject area. We propose a pipeline that fits broadly into two phases.

In Phase 1, we aim to construct a hierarchy of topics within a domain using various topic modelling techniques. To achieve this, we first parse the textbook knowledge models obtained using the processes described in previous research (see Section 2.1). This parsing process involves extracting the position of each section in the textbook hierarchy. Additionally, the following five key attributes are extracted for each section:

- **header**: the title of the textbook section (e.g. “Measures of Location”)
- **content**: the textual content associated with each section (e.g. “Visual summaries of data are excellent tools for obtaining preliminary impressions and insights...”)
- **concept names**: the DBpedia resources referenced in the textbook section (e.g. “Coefficient of variation”)

Table 1: Summary of related works

reference	contribution
Alpizar-Chacon and Sosnovsky [2, 4, 3, 5]	extracting knowledge models from textbooks
Alpizar-Chacon et al. [7]	intelligent online learning experience
Ramnarayan et al. [45]	textbooks as an effective source for topic modelling
Bullinaria and Levy [15]	effectiveness of cosine similarity in measuring semantic similarity
Guerra, Sosnovsky, and Brusilovsky [29]	effectiveness of LDA in linking topics from textbooks
Thaker, Brusilovsky, and He [53]	improved performance of Doc2Vec over LDA in textbook content linkage tasks
Pozzi, Alpizar-Chacon, and Sosnovsky [44]	use of BERT for automatic keyword extraction from textbooks
Lancichinetti et al. [33] and Li, Shang, and Yan [37]	effectiveness of LDA in classifying content into broad topic areas
Yang et al. [60]	fine-grained image classification

- **concept definitions**: the definitions associated with DBpedia resources referenced in the textbook section (e.g. "In probability theory and statistics, the coefficient of variation is a ...")
- **concept subjects**: the category of DBpedia resources referenced in the textbook section (e.g. "Statistical deviation and dispersion" for concept "Coefficient of variation").

Only concepts marked "core-domain" and "in-domain" are included. This is to avoid confusing the models with extraneous data. The definitions for domain specificity are provided in Section 2.1.

After parsing, the topic modelling methods are applied to the parsed textbook data and the performance of each technique is evaluated to address **RQ1** and choose an optimal technique. Using the best method, a comprehensive and unified data collection is generated from the textbook knowledge models and then converted into a normalised, labelled dataset.

Phase 2 involves the conversion of the labelled dataset into vector embeddings, and then the use of these vectors to train supervised deep learning models. **RQ2** explores the differences in performance seen in this phase when concept annotations are included in the input, while **RQ3** explores the differences in performance due to different dataset generation methods tested in Phase 1.

Figure 3 presents this methodology graphically, showing the various sub-parts of each phase and the prerequisite data for the pipeline.

3.1 Phase 1: Dataset Generation

3.1.1 Data

In this research, twelve textbooks from the statistics domain were used to develop our method:

- BookA: *Modern Mathematical Statistics with Applications*. Devore and Berk, 2011 [21]
- BookB: *Probability and statistics for engineers and scientists*. Walpole et al., 2010 [57]
- BookC: *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Dekking, 2005 [19]
- BookD: *Statistics for Non-Statisticians*. Madsen, 2011 [38]
- BookE: *A Concise Guide to Statistics*. Kaltenbach, 2011 [31]
- BookF: *Basic Concepts of Probability and Statistics in the Law*. Finkelstein, 2009 [27]
- BookG: *Statistics and Probability Theory: In Pursuit of Engineering Decision Support*. Faber, 2012 [26]
- BookH: *Statistics for Scientists and Engineers*. Shanmugam and Chattamvelli, 2015 [49]
- BookI: *Introductory Statistics for Business and Economics*. Ubøe, 2017 [54]
- BookJ: *Intuitive Introductory Statistics*. Wolfe and Schneider, 2017 [59]
- BookK: *OpenIntro Statistics*. Diez, Cetinkaya-Rundel, and Barr, 2019 [24]
- BookL: *Introductory Statistics with R*. Dalgaard, 2008 [17]

From this set, a base textbook is chosen and used as a foundation to hierarchically integrate the sections from other textbooks, such that the result is a table of contents where each entry points to groups of sections rather than a single section.

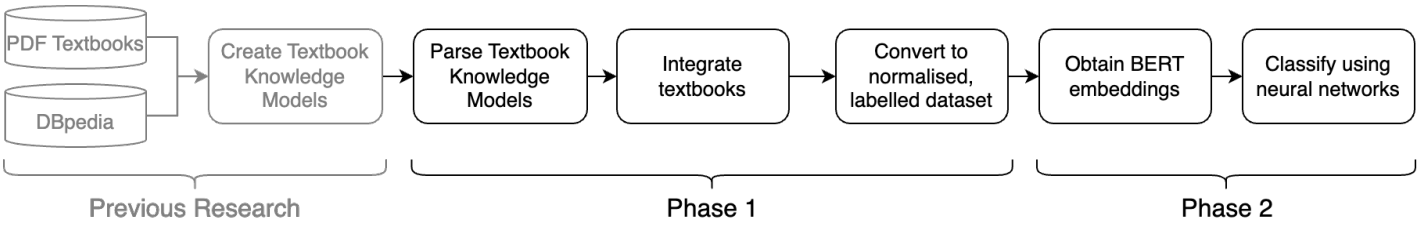
We chose BookA as the base textbook for two key reasons. Firstly, there exists an expert-generated mapping between the sections of BookA and BookB that can be used to evaluate the textbook integration methods (see Section 4.1). Additionally, BookA's sections seemed to provide a representation of the domain that possessed both the depth and breadth required to generate a fine-grained topic model. That is, it covered almost all the broader topics in the statistics domain while ensuring that each topic is explored effectively in the requisite subsections.

3.1.2 Textbook Integration Methods

Several methods were explored to integrate textbooks, each of which is described in the succeeding subsections. Each method is numbered and referred to alphabetically for simplicity.

Additionally, for each of the following methods, an "iterative" approach was tested, whereby, after each match, sections were iteratively updated to include the attributes of

Figure 3: Architecture for topic representation and classification



their newly matched section and the vectors for each section were recomputed. This modification was applied to each of the methods by allowing each method to take an additional parameter ("iterative") that can be set to true or false to control whether the incremental updating of the model should occur.

A. TF-IDF

A model[2] was used to convert each document into a vector of TF-IDF features. Vectors were then compared using cosine similarity. For each section in textbooks (excluding the base), the best matching section from the base textbook is found. The best match is determined by the pair having the highest cosine similarity. If the cosine similarity of this best match is above a certain threshold (a hyperparameter of the model), then the two sections can be considered a 'match'.

There are several hyperparameters to tune for this model.

- **Section Attributes:** The attributes of each section that are used to fit the model and compare similarity. For the set of section attributes A,

  $A = \{\text{header, content, concept name, concept definitions, concept subject}\}$,

  the set of choices for this parameter is the power set $\mathcal{P}(A)$, (i.e. the set of all possible subsets of A).

- **Threshold:** The minimum cosine similarity score for considering two sections a match. The values chosen were the fractional multiples of one-fifth, since taking smaller values would unnecessarily increase the processing time and taking larger values would not provide the granularity needed for an effective grid search. $\{0.2, 0.4, 0.6, 0.8\}$

- **Preprocessing:** The type of preprocessing to perform on the input text. $\{\text{none, lemmatization}\}$.

B. Doc2Vec

The Doc2Vec model[3] was used to learn document embeddings for the content of sections. Like the TF-IDF approach, a threshold parameter is set such that no matches below a given threshold will be accepted. A vector size parameter is set to control the dimensionality of feature vectors, and the minimum count parameter ignores all words with a total frequency less than the parameter value. The number of epochs was held constant at forty as this seemed to ensure the model was fit effectively but did not compromise performance.

- **Section Attribute:** Same as Method A. $\{\text{content, concept definitions}\}$.

- **Threshold:** Same as Method A. $\{0.2, 0.4, 0.5\}$

- **Vector Size:** Dimensionality of the embedding vectors. $\{50, 100, 200, 300\}$

- **Minimum Count:** Minimum frequency of words that should be included. $\{1, 5, 10, 20, 50\}$

C. Clustering

Another approach to measure the similarity of sections was to cluster sections based on their 'tags'. A tag could be a concept's name or subject, as defined in Section 2.1. A similar approach, whereby the content of textbooks is represented at the level of domain concepts rather than using the literal content is explored in Thaker, Brusilovsky, and He [53]. In this research, K-means clustering is used.

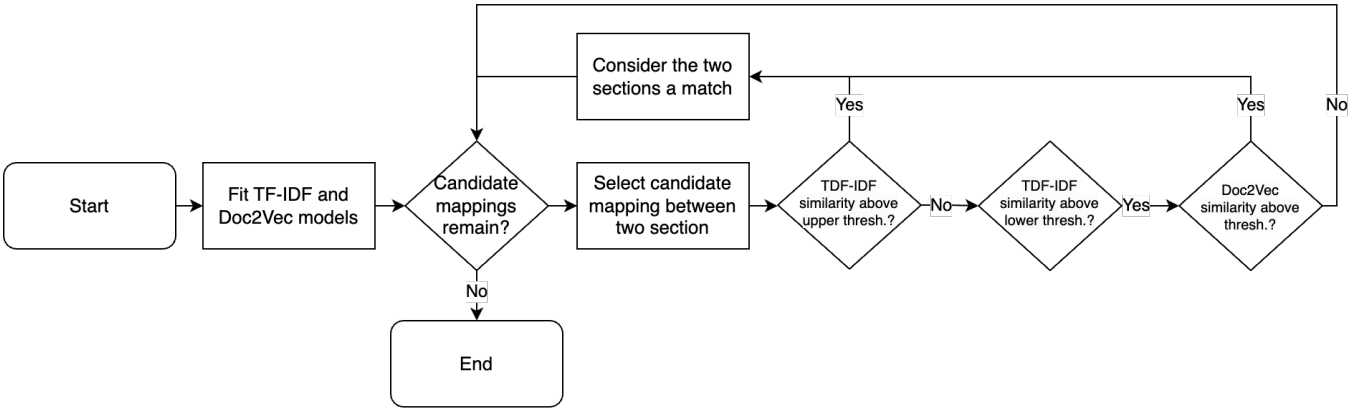
Each section is represented as a binary vector, where each element indicates the presence or absence of a particular tag. Sections are clustered based on these vectors. The similarity score between two sections is binary and determined by whether two sections are in the same cluster. In the case that multiple attributes are used, the weighted average of these binary outcomes is taken, to achieve a continuous similarity score that is tested against a minimum threshold to ensure that sections can be considered similar.

- **Section Attributes:** The attribute(s) that the sections are clustered on. $\mathcal{P}$ ($\{\text{concept name, concept category}\}$).

- **Number of Clusters:** A hyperparameter of the K-means clustering algorithm. When clustering is performed multiple times (i.e. multiple section attributes are used), the

[2] TfidfVectorizer from scikit-learn [42]
[3] from Gensim [46]

Figure 4: Flowchart describing Method E



number of clusters parameter must be set as many times as a clustering algorithm is run (e.g. for two section attributes, use two values of the number of clusters parameter). $\{40, 60, 80, 100, 120\}$

- **Threshold:** A minimum similarity score to consider two sections a match. Only used for multiple section attributes, as described above. $\{0.2, 0.4, 0.6\}$
- **Weights:** How much weight to give each attribute when taking the weighted average of similarity scores (for multiple attributes only). $\{X \in \mathcal{P}(\{0.2, 0.4, 0.6, 0.8\}) \mid |X| = 2\}$

D. TF-IDF & Clustering Ensemble Averaging

It is known that ensemble algorithms perform better than their singular counterparts [41]. This is based on the common societal practice of 'seeking a second opinion' [43]. Thus, Methods A & C were combined to form an ensemble model. The parameters of each of these methods are used in the same way in this ensemble approach.

E. TF-IDF & Doc2Vec Iterative Hybrid

Similarly, it was observed that the Doc2Vec and TF-IDF models have different strengths and weaknesses. For example, TF-IDF had better precision while Doc2Vec had better recall. TF-IDF can observe exact matches in attributes of the section that are less likely to have synonyms (e.g. the header or concepts), while Doc2Vec can observe similarities in the textual content, which might be worded in any number of ways. Thus, a hybrid approach was implemented whereby the models are combined iteratively. Based on cosine similarity thresholds, the TF-IDF model first classifies some matches between sections as 'definitive', while maintaining a record of some 'uncertain' matches. Simultaneously a Doc2Vec model is fit. Matches that are within the uncertain thresholds for the first model are then checked against the Doc2Vec model to see if they have a score above the Doc2Vec threshold. These matches are added to the resulting integration of sections.

This approach uses the same parameters that are used in Methods A & B. Additionally, there is a new parameter, "Uncertain Threshold", that provides a minimum TF-IDF similarity score value for a match to be included. If a match scores above this value, but below the TF-IDF standard threshold, the match's similarity score in the Doc2Vec model will be checked to see whether it would be considered a match within the Doc2Vec framework, essentially given the potential match a 'second chance'. This process is described by the flowchart in Figure 4.

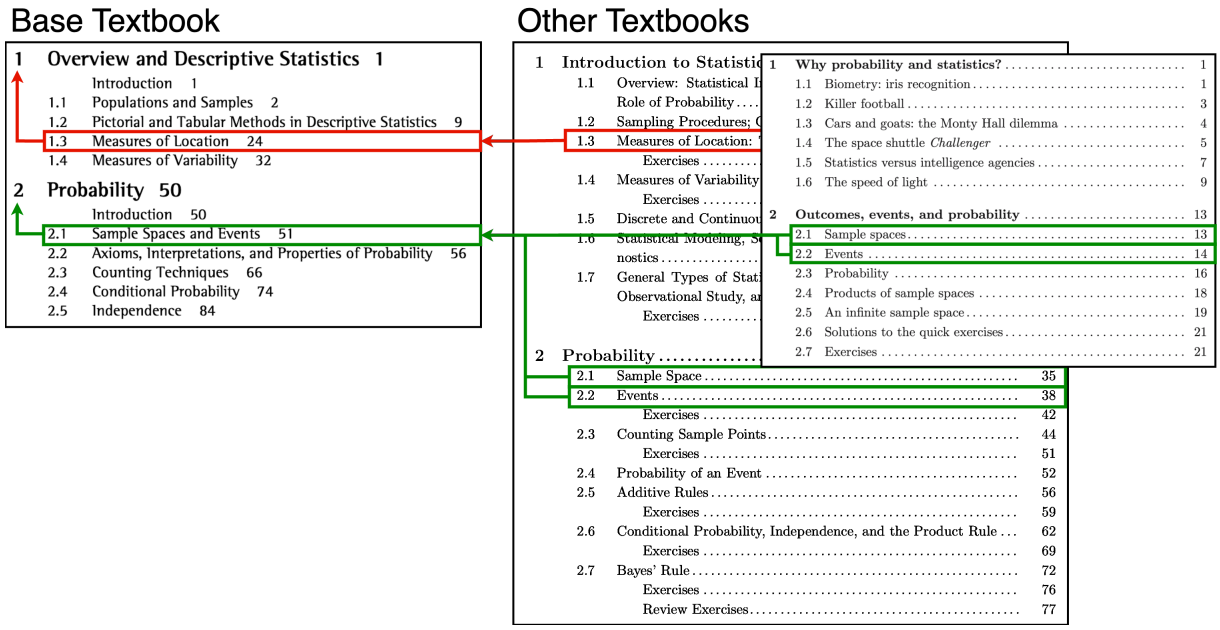
3.1.3 Converting Integrated Textbook to Labelled Dataset

Once the textbooks are integrated using the optimal method, it is necessary to convert the newly created hierarchy of textbook sections into a normalised, labelled dataset that can be used for supervised machine learning. We perform this conversion by taking each section (and all their attributes) to be a single instance in our dataset. Then, the class label is set to be the section's largest ancestor.

Consider Figure 5 as an example of this approach. In the diagram, there are seven highlighted sections across three textbooks. However, these fit into only two class labels ("Overview and Descriptive Statistics" and "Probability") since these are the top-level sections from the base textbook. This approach is taken to ensure that each class has sufficient membership.

From the previous methods, the optimal method is used to generate two hierarchical topic models to convert to labelled datasets. The first, known hereafter as "small", is generated by using the optimal method to integrate the same two textbooks that were used to evaluate the textbook integration methods (`BookA` and `BookB`). The second dataset ("large") is generated by using the optimal method to integrate all twelve textbooks. Additionally, the expert's man-

Figure 5: Example conversion from integrated textbook's hierarchy of sections to normalised labelled dataset



ual hierarchy of topics, described in Section 4.1, is used to generate a third "expert" dataset.

3.2 Phase 2: Topic Classification

Once the three new datasets have been generated, they are used to train an advanced language model for the classification task.

First, vector embeddings for each textbook section are obtained using BERT. BERT accepts special tokens ([CLS] and [SEP]) which can be used to provide the model with two different features of the input sequence, so we test two approaches. In the first approach, concept names are prepended to the section's content, while the second approach simply uses the section's content. Consider the following example BERT tokenisation input where concepts are prepended to the input sequence for a section titled "Measures of Variability" :

> [CLS] Degrees of freedom Population Parameter Range Standard deviation Variance Population [SEP] Sample variability plays an important role in data analysis. Process and product variability is a fact of life in engineering and scientific systems: The control or reduction of process variability is often a source ... [SEP]

This approach is similar to the approach to BERT input encoding taken in Pozzi, Alpizar-Chacon, and Sosnovsky [44]: concept names are used to summarise the section's textual content.

We propose that this addition allows the model to differentiate between the two different kinds of information and learn how concepts are described by the content, enhancing its ability to understand and represent the relationship between these two elements in the text. This proposal is explored as a part of **RQ2**.

The input text is then tokenized and truncated or padded to ensure that the length of each input is exactly 512 tokens. To convert input tokens to fixed-size tensors, the inputs must be the same length. One approach is to pad all sequences with a special token (usually 0) so that each sequence is the same length as the longest sequence in the batch. However, due to the Transformer network architecture of BERT, its time complexity involves a quadratic dependency of the sequence length [56]. Therefore, to avoid degraded performance, a token limit of 512 is set and all sequences with more tokens are truncated. It is proposed that the content of a textbook section might be effectively summarised by the first 512 tokens since textbooks typically include an all-encompassing introduction at the beginning of sections and subsections. Thus, it is assumed that minimal information will be lost through the truncation process.

Once the input is tokenised, it is passed to the pre-trained DistilBERT model[4], which converts the text into a high-dimensional vector representing various linguistic features. In this research, no fine-tuning is performed on top of the pre-trained BERT model, since this would require a labelled dataset. While our dataset does have labels, these labels will be used later to train RNNs for classification. Fine-tuning the BERT model using these labels will result in data leakage, i.e. the training data for our RNNs will contain information about the targets.

[4] from the Hugging Face transformers library [58]

These vectors are then used to train RNNs to classify new sections according to their topic. It's important to note that the class label originates from the largest ancestor of the section in the integrated textbook. In other words, if the section comes from the base textbook, the class label is the top-level section that it belongs to. If the section comes from the other textbooks, its class label is the top-level section that its base textbook match belongs to. This approach is taken to ensure that each class has enough members.

Additionally, there are parameters available to be chosen for RNNs:

- **Batch Size** (N)**:** the number of training examples used in one iteration of model training (i.e. used to compute the gradient and update the model parameters).
- **Dropout Rate** (p)**:** the probability of an input unit being dropped or set to zero during training. This helps prevent overfitting by forcing the network to learn redundant representations.
- **Units** (u)**:** the number of units or neurons in a layer.

To select the optimal parameters for the neural network, a grid search with 5-fold cross-validation was performed. As part of the grid search, we also varied the model type between RNN and LSTM.

Accuracy is chosen as the key metric to optimise and evaluate models. This is because it is a direct reflection of the number of correct predictions and, particularly in the case of multi-class classification, it is simple and easy to interpret.

For each dataset, we performed the grid search twice. This was to test the model's performance when concepts were both included and omitted from the input text, allowing us to explore **RQ2**.

4 Results & Evaluation

This section presents the results that were seen when testing our proposed methodology. Additionally, we attempt to answer our research questions with a greater level of certainty by using statistical hypothesis testing. For all statistical tests, the level of significance is $\alpha = 0.05$. Furthermore, the p-values are adjusted using the Bonferroni correction to avoid the multiple comparisons problem.

4.1 Phase 1: Dataset Generation

To evaluate the performance of the textbook integration task, we used a "ground truth" produced by three experts from Utrecht University for previous research [4]. This data consists of a manual mapping between the chapters, sections, and subsections of BookA to those of BookB. Experts had the flexibility to link textbook sections at various levels within the table of contents, allowing for many-to-many mappings. They were also asked to specify the strength of the mappings on a scale from one to three, however, this additional information was not used in this research.

It is important to note that, given the expert ground truth only exists for mappings between two of the textbooks, it is impossible to evaluate the textbook linkage performance on the entire dataset. One possible implication is that the best textbook integration method is selected based on the integration of two textbooks, and such a method might not be able to be effectively extended to the integration of twelve textbooks.

The standard precision and recall metrics were used to compare the performance of the different textbook linking algorithms. In this context, precision is the proportion of the mappings identified by the algorithm that the experts also identify. Recall is the proportion of the mappings identified by the experts that the algorithm also identifies.

While both metrics are important, it can be argued that precision is the more important metric of the pair in the domain of online education. While both false positives and false negatives are undesirable, in many educational contexts, the cost of a false positive (e.g. wrongly associating two topics) can have greater cost than a false negative (e.g. failing to identify a similarity between two topics). Precision focuses on minimising false positives.

However, it is important to note that this does not mean recall is unimportant, especially since the downstream uses of this research are not yet specified in detail. A balanced approach that considers both precision and recall is often the best strategy, especially in scenarios where missing out on key information (low recall) could be as detrimental as providing incorrect information (low precision).

Thus, it was decided to use the F_β score for evaluation, which is the weighted harmonic mean of precision and recall. The β parameter represents the importance of recall over precision, such that $\beta = 0.5$ means that we consider recall to be half as important as precision [55]. The formula for the F_β score is shown in Equation 3.

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \tag{3}$$

For each integration strategy, we performed a grid search to find the optimal parameter choices. This led to 984 different combinations of parameters and algorithms. Table 2 shows the results of the best-performing parameter combinations for each textbook integration method.

Out of the individual models, Method A and Method B seem to be the best performing. Given that Method E improves on the results of these models, with particularly good precision, the approach of combining Methods A & B seems to be successful.

It should be noted that the iterative approach described previously did not yield the best results for any of the strategies, but it did result in a significant increase in computation time.

Table 2: Results of optimal model for each textbook integration method

method	TP	FP	FN	precision	recall	F_1	F_β
A	48	33	94	0.593	0.338	0.430	0.515
B	56	42	86	0.571	**0.394**	**0.467**	0.524
C	4	150	138	0.026	0.028	0.027	0.026
D	44	51	98	0.463	0.310	0.371	0.421
E	45	9	97	**0.833**	0.317	0.459	**0.628**

To determine whether Method E has a statistically significant advantage over the other methods according to the F_β score (and thus answer **RQ1**), a Mann–Whitney U test was performed with the following null hypothesis:

H_0: The distribution underlying the F_β scores for Method E is not stochastically greater than the distributions underlying the F_β scores for the other methods.

Table 3: Results of Mann–Whitney U test for **RQ1**

method	μ	σ	U	adj. p
A	0.363	0.120	9981.0	***
B	0.276	0.124	25491.5	***
C	0.008	0.008	33120.0	***
D	0.136	0.083	48335.0	***
E	0.484	0.054	–	–

*** indicates $p \leq 0.001$

The Mann–Whitney U test is the optimal statistical test in this scenario since the samples are not paired and the Shapiro–Wilk test shows that the data is not normally distributed. The results of this test are shown in Table 3. The results are statistically significant, allowing us to reject the null hypothesis. Thus, Method E is statistically more performant than the other methods, answering **RQ1**.

Thus, Method E (TF-IDF & Doc2Vec Iterative Hybrid model) is used in the next phase. The optimal parameters and attributes to be used with this method are as follows:

- **TF-IDF Section Attributes:** {content, concept names}
- **TF-IDF Upper Threshold:** 0.6
- **TF-IDF Lower Threshold:** 0.3
- **Doc2Vec Section Attribute:** content
- **Doc2Vec Threshold:** 0.6
- **Doc2Vec Vector Size:** 100
- **Doc2Vec Minimum Count:** 1

This Method E was then used to generate two hierarchical topic models ("small" and "large", as described in Section 3.1.3), which are then converted to labelled datasets. Additionally, the expert's manual hierarchy of topics is also converted to a labelled dataset. Table 4 displays some features of these datasets.

Table 4: Features of generated datasets

dataset	instances	topic labels
small	354	32
large	2368	329
expert	216	14

4.2 Phase 2: Topic Classification

As outlined in Section 3.1.3, three different datasets (generated using the process described in Phase 1) were used to train and evaluate the neural networks in this stage. Additionally, the classification models were tested both with and without concepts prepended to the sequence. The optimal models that were found in each grid search are displayed in Table 5.

Table 5: Selected parameters for each dataset

dataset	concepts	model	N	p	u
expert	true	SimpleRNN	128	0.4	125
expert	false	LSTM	128	0.4	200
small	true	SimpleRNN	32	0.9	100
small	false	SimpleRNN	64	0.9	100
large	true	SimpleRNN	64	0.9	125
large	false	SimpleRNN	64	0.9	100

For each of these optimal models, we compared the performance of the model to that of a baseline, chosen to be BERTopic[5][28]. BERTopic is a topic modelling framework based on BERT that has several capabilities, including out-of-the-box topic classification. It was chosen as a baseline due to its ease of use as well as its robust performance across a variety of tasks.

The performance of the optimal models for each dataset, as well as the baseline performance, are displayed in Table 6. The results presented are the average of 5-fold cross-validation. In every metric, our models perform better than the BERTopic baseline.

Additionally, across the expert and large datasets, our model performs better when concepts are prepended to the

[5] https://github.com/MaartenGr/BERTopic

Table 6: Test performance of the optimal model for each dataset

dataset	concepts	accuracy		precision		F_1	
		model	baseline	model	baseline	model	baseline
expert	true	**0.59**	0.13	**0.67**	0.07	0.61	0.08
expert	false	0.41	0.13	0.50	0.09	0.44	0.09
small	true	0.35	0.03	0.48	0.01	0.53	0.02
small	false	0.45	0.03	0.64	0.01	**0.62**	0.02
large	true	0.26	0.00	0.56	0.00	0.40	0.00
large	false	0.24	0.00	0.57	0.00	0.37	0.00

Recall is excluded because weighted-average recall is equivalent to accuracy for multiclass classification. Weighted-average precision is used.

input text. Statistical tests, using the following null hypothesis, were performed to confirm this and thus answer **RQ2**.

H_0: The distribution of accuracy scores is not stochastically greater when including concepts in the input embeddings.

The Shapiro–Wilk test confirmed that the data was normally distributed, allowing the use of the paired samples t-test. In this test, the paired samples are the accuracy scores achieved under the same conditions both with and without the inclusion of concepts. The results, displayed in Table 7, are statistically significant for the expert and large datasets. Therefore, for these datasets, we can reject the null hypothesis that there is no improvement in accuracy when concepts are included in the input embeddings.

Table 7: Results of paired samples t-test for **RQ2**

dataset	μ	σ	t	adj. p
expert	0.015	0.026	5.801	***
small	0.003	0.017	1.599	0.170
large	0.006	0.008	4.117	***

*** indicates $p \leq 0.001$

Additionally, the expert dataset performs better than the generated datasets and the large, generated dataset performs particularly badly. This might be due to the large number of classes introduced by integrating all twelve textbooks simultaneously.

Statistical tests were performed to confirm whether the expert dataset results in the best performance across all metrics. The data is not normally distributed (as confirmed by the Shapiro–Wilk test), therefore, the Mann–Whitney U test was used with the following null hypothesis:

H_0: The distribution of accuracy scores for classification using the expert dataset is not stochastically greater than the metrics achieved on the large and small datasets.

The results are displayed in Table 8. All of the adjusted p-values are significant, allowing us to reject the null hypothesis. Therefore, answering **RQ3**, we can say with some certainty that the manually-generated expert dataset performs better at the topic classification task than the datasets that were automatically generated using our optimal content linkage method.

5 Discussion & Limitations

Overall, this research has shown that extracting fine-grained, domain-specific topic models from textbooks is effective. While there are obvious limitations in many parts of our proposed architecture, these could be resolved in future research, with tremendous implications for the facilitation of tasks such as learning object classification, automatic index creation, and content personalisation.

When comparing the textbook integration techniques used in Phase 1 for **RQ1**, it is interesting to note that a combination of two individual approaches is the best performing (i.e. the combination of Methods A & B resulting in Method E). This is in line with expectations that ensemble learning results in performance improvements. However, it is important to note that combining two models to form an ensemble model results in significant increases in computation time. Further research might explore how these methods might be optimised or combined more efficiently.

Furthermore, when converting the generated hierarchy of topics into a labelled dataset (described in Section 3.1.3), it was decided to take a section's topic label as its largest ancestor in the hierarchy. This likely results in a less fine-grained model than if a more precise topic label were taken (e.g. from the second or third largest ancestor). While this results in additional complexities, such as low class membership counts, further research might explore how these can be mitigated to create topic models that possess even finer granularity.

For **RQ2**, the inclusion of concept annotations significantly improved model performance for the large and expert datasets. It should also be noted that this performance is using a pre-trained DistilBERT implementation without fine-tuning. Perhaps with more time or resources, a model that is more customised to our use case could be fine-tuned. An

Table 8: Results of Mann–Whitney U test for **RQ3**

dataset	with concepts				without concepts			
	μ	σ	U	adj. p	μ	σ	U	adj. p
large	0.25	0.03	2304.0	***	0.24	0.03	2304.0	***
small	0.30	0.01	9216.0	***	0.30	0.02	9216.0	***
expert	0.52	0.03	–	–	0.51	0.03	–	–

*** indicates $p \leq 0.001$

alternative approach that might also be explored in future research is the use of a fine-tuned BERT model for classification instead of using RNNs trained on BERT embeddings.

Additionally, the way that we have used the BERT special tokens is not entirely consistent with the intended usage for the pre-trained BERT and DistilBERT models. Specifically, the `[SEP]` token is intended to separate the tokens into two sentences for the NSP task. However, when we prepend concepts to the input, the first 'sentence' in our input is not a true sentence but rather a list of words that do not make sense when strung together as a sentence (e.g. "Degrees of freedom Population Parameter Range Standard deviation Variance Population"). With more time and resources, a custom implementation of BERT might be pre-trained so that the first sentence of the input is more syntactically alike to our first sentences.

Furthermore, due to constraints in computing time and resources, only the first 512 tokens of each input sequence were passed to the BERT. It might be the case that this small part of the text is inadequate at capturing the full meaning of each textbook section. Therefore, performance benefits might be seen if the maximum number of tokens for the input sequence is increased, or if longer sequences are split to form a larger dataset.

Also, the investigation into **RQ3** revealed that the quality of datasets plays a crucial role, with higher quality datasets leading to improved performance of machine learning models. The strong performance of the expert dataset in the topic classification tasks allows us to assume that if we improve the textbook integration in Phase 1, we might be able to improve the performance of topic classification on the generated dataset in Phase 2.

Additionally, in this research, the Python programming language has been used to allow for the fast development of a proof of concept. Python was an effective choice in achieving this goal, due to its simplicity, readability, and the vast availability of libraries and frameworks to streamline the development process. However, Python's interpreted nature can make it less efficient for fast or high-performance machine learning tasks compared to languages that are compiled and optimised for speed. Additionally, Python is limited in its multi-threaded execution capabilities, meaning there are limited gains to be achieved by executing independent code in parallel. Future research might focus more on increasing the speed and efficiency of the code so that more powerful models can be used, or the models can be applied to a larger dataset.

There were minimal ethical impacts concerning this research since the research does not involve humans and there are no immediately obvious negative uses or consequences of this research. The data being used does not contain any Personal Identifiable Information. While a general ethical consideration for AI models is that they may inadvertently learn and perpetuate biases in their training data, this concern is mitigated by the fact that the data consists of scientific textbooks, which are generally considered to be very unbiased sources of information.

Our research marks a significant development towards personalised learning, where educational applications can be customised to meet the unique needs and learning paths of individuals. Accurately constructing domain models from textbooks provides a robust framework for various tasks that can improve the overall educational experience. Therefore, this research not only contributes to the academic discourse on AI's role in education but also paves the way for practical applications in personalised learning environments.

6 Conclusion

In this research, we have presented a novel architecture for the classification of topics in educational content, focusing on the extraction of domain-specific and fine-grained topic models from textbooks. Our work takes a significant step forward in applying artificial intelligence to enhance the digital learning experience, addressing the challenge of accurately classifying and linking educational content across various domains. The performance of this architecture is not strong enough for use in production applications; however, the initial results give promise that further research and investments in computing power might yield a pipeline that can generate more correct data and classify new data more accurately.

The implementation of the first phase of our architecture revealed the significant complexity inherent in developing an efficient and performant content linkage tool. Despite limitations due to the availability of expert mappings for only a subset of textbooks, our comparison of textbook

linking algorithms provided insightful data on the effectiveness of these methods. Our results indicate that a TF-IDF & Doc2Vec Iterative Hybrid model, showed reasonably high levels of precision, suggesting a promising approach for integrating textbook content.

In Phase 2, our RNN models trained on BERT embeddings demonstrated improved performance across all metrics compared to the BERTopic baseline, particularly when concept annotations were prepended to the input text. This suggests that the inclusion of domain-specific concepts significantly enhances model accuracy, especially for the expert and large datasets, validating our approach to model training.

Although challenges remain, such as the moderate performance of our automatic content linkage method compared to expert manual efforts, our approach to topic classification shows promise. Our findings underscore the importance of using high data quality to optimise the performance of advanced machine learning models in classifying textbook content. The reasonably strong performance of topic classification on the expert dataset illustrates that, if accurately generated, topic models constructed from textbooks can accurately reflect the depth and breadth of subjects and thus result in a strong performance when used as inputs into domain-specific and fine-grained topic classification.

Our research has demonstrated the potential of leveraging the intricate knowledge embedded in textbooks in conjunction with advanced AI and NLP techniques in revolutionising the online learning experience. By adopting a methodical approach to extract and utilise fine-grained, domain-specific topic models, we have uncovered insights that not only answer our primary research question regarding the efficacy of machine learning methods in this context but also pave the way for future innovations in educational technology, such as improved searchability, personalization, and recommendation systems in online learning systems.

Future research must continue to explore and refine AI-based educational tools, focusing on overcoming current limitations and enhancing the efficacy of topic models. Constraints relating to computing time and resources are the key limitations that should be explored in future research. By harnessing the power of domain-specific, fine-grained topic modelling, we move closer to realising the vision of a more efficient, effective, and accessible educational system for learners.

The code for this research is available in a public GitHub repository[6].

References

[1] Micheal Olalekan Ajinaja et al. "Semantic similarity measure for topic modeling using latent Dirichlet allocation and collapsed Gibbs sampling". In: *Iran Journal of Computer Science* 6.1 (Mar. 2023), pp. 81–94. ISSN: 2520-8446. DOI: 10.1007/s42044-022-00124-7.

[2] Isaac Alpizar-Chacon and Sergey Sosnovsky. "Expanding the Web of Knowledge: One Textbook at a Time". In: *Proceedings of the 30th ACM Conference on Hypertext and Social Media*. HT '19. Hof, Germany: Association for Computing Machinery, 2019, pp. 9–18. ISBN: 9781450368858. DOI: 10.1145/3342220.3343671.

[3] Isaac Alpizar-Chacon and Sergey Sosnovsky. "Knowledge models from PDF textbooks". In: *New Review of Hypermedia and Multimedia* 27.1-2 (2021), pp. 128–176. DOI: 10.1080/13614568.2021.1889692.

[4] Isaac Alpizar-Chacon and Sergey Sosnovsky. "Order out of Chaos: Construction of Knowledge Models from PDF Textbooks". In: *Proceedings of the ACM Symposium on Document Engineering 2020*. DocEng '20. Virtual Event, CA, USA: Association for Computing Machinery, 2020. ISBN: 9781450380003. DOI: 10.1145/3395027.3419585.

[5] Isaac Alpizar-Chacon and Sergey Sosnovsky. "What's in an Index: Extracting Domain-Specific Knowledge Graphs from Textbooks". In: *Proceedings of the ACM Web Conference 2022*. WWW '22. Virtual Event, Lyon, France: Association for Computing Machinery, 2022, pp. 966–976. ISBN: 9781450390965. DOI: 10.1145/3485447.3512140.

[6] Isaac Alpizar-Chacon, Sergey Sosnovsky, and Peter Brusilovsky. "Measuring the Quality of Domain Models Extracted from Textbooks with Learning Curves Analysis". In: *Artificial Intelligence in Education*. Ed. by Ning Wang et al. Cham: Springer Nature Switzerland, 2023, pp. 804–809. ISBN: 978-3-031-36272-9.

[7] Isaac Alpizar-Chacon et al. "Transformation of PDF Textbooks into Intelligent Educational Resources". In: *iTextbooks 2020*. Proceedings of the Second International Workshop on Intelligent Textbooks 2020. July 2020.

[8] Dimo Angelov. *Top2Vec: Distributed Representations of Topics*. 2020. arXiv: 2008.09470 [cs.CL].

[9] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181.

[10] D. Benta, G. Bologa, and I. Dzitac. "E-learning Platforms in Higher Education. Case Study". In: *Procedia Computer Science* 31 (2014). 2nd International Conference on Information Technology and Quantitative Management, ITQM 2014, pp. 1170–1176. ISSN: 1877-0509. DOI: 10.1016/j.procs.2014.05.373.

[11] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[12] Elia Bruni, Nam Khanh Tran, and Marco Baroni. "Multimodal distributional semantics". In: *Journal of Artificial Intelligence Research* 49 (Jan. 2014), pp. 1–47. DOI: 10.1613/jair.4135.

[6] https://github.com/CobySim01/textbook-topic-analysis

[13] Peter Brusilovsky. "Adaptive Navigation Support". In: *The Adaptive Web: Methods and Strategies of Web Personalization*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 263–290. ISBN: 978-3-540-72079-9. DOI: 10.1007/978-3-540-72079-9_8.

[14] Peter Brusilovsky and John Eklund. "A Study of User Model Based Link Annotation in Educational Hypermedia". In: *Journal of Universal Computer Science* 4 (June 1998).

[15] John A. Bullinaria and Joseph P. Levy. "Extracting semantic representations from word co-occurrence statistics: A computational study". In: *Behavior Research Methods* 39.3 (Aug. 2007), pp. 510–526. ISSN: 1554-3528. DOI: 10.3758/BF03193020.

[16] Rob Churchill and Lisa Singh. "The Evolution of Topic Modeling". In: *ACM Comput. Surv.* 54.10s (Nov. 2022). ISSN: 0360-0300. DOI: 10.1145/3507900.

[17] Peter Dalgaard. *Introductory Statistics with R*. Second. Statistics and Computing. New York: Springer, 2008. ISBN: 978-0-387-79053-4. DOI: 10.1007/978-0-387-79054-1.

[18] Scott Deerwester et al. "Indexing by latent semantic analysis". In: *Journal of the American Society for Information Science* 41.6 (1990), pp. 391–407. DOI: https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[19] F.M. Dekking. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Springer Texts in Statistics. Springer, 2005. ISBN: 9781852338961.

[20] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[21] J.L. Devore and K.N. Berk. *Modern Mathematical Statistics with Applications*. Springer Texts in Statistics. Springer New York, 2011. ISBN: 9781461403906.

[22] Christo Dichev and Darina Dicheva. "View-Based Semantic Search and Browsing". In: *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. WI '06. USA: IEEE Computer Society, 2006, pp. 919–925. ISBN: 0769527477. DOI: 10.1109/WI.2006.187.

[23] Oxford English Dictionary. In: Oxford University Press, Dec. 2023. DOI: 10.1093/OED/8858470422.

[24] D.M. Diez, M. Cetinkaya-Rundel, and C.D. Barr. *OpenIntro Statistics*. Open textbook library. OpenIntro, Incorporated, 2019. ISBN: 9781943450084.

[25] Barbara Ericson. "An Analysis of Interactive Feature Use in Two Ebooks". In: *Proceedings of the First Workshop on Intelligent Textbooks co-located with 20th International Conference on Artificial Intelligence in Education (AIED 2019), Chicago, IL, USA, June 25, 2019*. Ed. by Sergey A. Sosnovsky et al. Vol. 2384. CEUR Workshop Proceedings. CEUR-WS.org, 2019, pp. 4–17. URL: http://ceur-ws.org/Vol-2384/paper01.pdf.

[26] Michael Havbro Faber. *Statistics and Probability Theory: In Pursuit of Engineering Decision Support*. English. Topics in Safety, Risk, Reliability and Quality. Springer Publishing Company, 2012. ISBN: 978-9400740556.

[27] Michael Finkelstein. *Basic Concepts of Probability and Statistics in the Law*. Jan. 2009. ISBN: 978-0-387-87500-2. DOI: 10.1007/b105519.

[28] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022).

[29] Julio Guerra, Sergey Sosnovsky, and Peter Brusilovsky. "When One Textbook Is Not Enough: Linking Multiple Textbooks Using Probabilistic Topic Models". In: *Scaling up Learning for Sustained Impact*. Ed. by Davinia Hernández-Leo et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 125–138. ISBN: 978-3-642-40814-4. DOI: 10.1007/978-3-642-40814-4_11.

[30] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[31] H.M. Kaltenbach. *A Concise Guide to Statistics*. SpringerBriefs in Statistics. Springer Berlin Heidelberg, 2011. ISBN: 9783642235023.

[32] Krishna Kumar. "Textbooks and Educational Culture". In: *Economic and Political Weekly* 21.30 (1986), pp. 1309–1311. ISSN: 00129976, 23498846.

[33] Andrea Lancichinetti et al. *A high-reproducibility and high-accuracy method for automated topic classification*. 2014. arXiv: 1402.0422 [stat.ML].

[34] Quoc V. Le and Tomas Mikolov. *Distributed Representations of Sentences and Documents*. 2014. arXiv: 1405.4053 [cs.CL].

[35] Kathy Lee et al. "Twitter Trending Topic Classification". In: *2011 IEEE 11th International Conference on Data Mining Workshops*. 2011, pp. 251–258. DOI: 10.1109/ICDMW.2011.171.

[36] Jens Lehmann et al. "DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia". In: *Semantic Web* 6 (2015). 2, pp. 167–195. DOI: 10.3233/SW-140134.

[37] Zhenzhong Li, Wenqian Shang, and Menghan Yan. "News text classification model based on topic model". In: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. 2016, pp. 1–5. DOI: 10.1109/ICIS.2016.7550929.

[38] Birger Madsen. *Statistics for Non-Statisticians*. Springer Berlin Heidelberg, 2011. ISBN: 9783642176562.

[39] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

[40] Saif M. Mohammad and Graeme Hirst. *Distributional Measures of Semantic Distance: A Survey*. 2012. arXiv: 1203.1858 [cs.CL].

[41] David Opitz and Richard Maclin. "Popular Ensemble Methods: An Empirical Study". In: *Journal of Artificial Intelligence Research* 11 (Aug. 1999), pp. 169–198. DOI: 10.1613/jair.614.

[42] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[43] Robi Polikar. “Ensemble based systems in decision making”. In: *IEEE Circuits and Systems Magazine* 6.3 (2006), pp. 21–45. DOI: 10.1109/MCAS.2006.1688199.

[44] Lorenzo Pozzi, Isaac Alpizar-Chacon, and Sergey Sosnovsky. “Harnessing Textbooks for High-Quality Labeled Data: An Approach to Automatic Keyword Extraction”. In: *iTextbooks 2023*. Fifth Workshop on Intelligent Textbooks. July 2023.

[45] P Ramnarayan et al. “ISABEL: a web-based differential diagnostic aid for paediatrics: results from an initial performance evaluation”. en. In: *Arch Dis Child* 88.5 (May 2003), pp. 408–413.

[46] Radim Řehůřek and Petr Sojka. “Software Framework for Topic Modelling with Large Corpora”. English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[47] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: 1910.01108 [cs.CL].

[48] Satoshi Sekine. “The domain dependence of parsing”. In: *Fifth Conference on Applied Natural Language Processing*. 1997, pp. 96–102.

[49] R. Shanmugam and R. Chattamvelli. *Statistics for Scientists and Engineers*. Wiley, 2015. ISBN: 9781119047186.

[50] Sergey Sosnovsky. “Math-Bridge: Closing Gaps in European Remedial Mathematics with Technology-Enhanced Learning”. In: *Mit Werkzeugen Mathematik und Stochastik lernen – Using Tools for Learning Mathematics and Statistics*. Ed. by Thomas Wassong et al. Wiesbaden: Springer Fachmedien Wiesbaden, 2014, pp. 437–451. ISBN: 978-3-658-03104-6. DOI: 10.1007/978-3-658-03104-6_31.

[51] Karen Spärck Jones. “A Statistical Interpretation of Term Specificity and its Application in Retrieval”. In: *Journal of Documentation* 28.1 (Jan. 1972), pp. 11–21. ISSN: 0022-0418. DOI: 10.1108/eb026526.

[52] Ralf C. Staudemeyer and Eric Rothstein Morris. *Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks*. 2019. arXiv: 1909.09586 [cs.NE].

[53] Khushboo Maulikmihir Thaker, Peter Brusilovsky, and Daqing He. “Concept Enhanced Content Representation for Linking Educational Resources”. In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. 2018, pp. 413–420. DOI: 10.1109/WI.2018.00-59.

[54] Jan Ubøe. *Introductory Statistics for Business and Economics*. Springer, 2017.

[55] C.J. Van Rijsbergen. *Information Retrieval*. Butterworths, 1979. ISBN: 9780408709293.

[56] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[57] Ronald E Walpole et al. *Probability and statistics for engineers and scientists*. 9th ed. Upper Saddle River, NJ: Pearson, Dec. 2010.

[58] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[59] Douglas A. Wolfe and Grant Schneider. *Intuitive Introductory Statistics*. Springer Texts in Statistics. Cham: Springer, 2017. ISBN: 978-3-319-56070-0. DOI: 10.1007/978-3-319-56072-4.

[60] Ze Yang et al. “Learning to Navigate for Fine-grained Classification”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.