Topic Analysis of Textbooks Using AI

\/\/\

Advances in the field of artificial intelligence have opened new horizons for personalized learning. This research project focuses on the application of AI in identifying and categorising topics within educational content.

\/\/\

We aim to construct domain-dependent and fine-grained topic models of a very particular domain – statistics in our case. To our knowledge, there has been little research in the extraction of closely related and overlapping topics from textual data.

\/\/\

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.

\/\/\

First, I'll review some previous research. Then, I'll present our proposed methodology and evaluate it. I'll finish up with a discussion and some comments about future research.

Previous Research

First, let's discuss some of the previous research in this space.

Modelling of Textbooks

Alpizar-Chacon and Sosnovsky have proposed an intelligent and personalised online learning experience, whereby a learner can access enriched semantic information that is external to the textbook itself. This method of presenting the textbook is backed by rich knowledge models, developed by parsing PDF textbooks, and encoding them according to the Text Encoding Initiative guidelines – a framework that encompasses around 500 distinct textual and conceptual elements. The textbooks' glossary and index are then annotated with links to DBpedia – a linked open data cloud database that extracts structured content from Wikipedia. These links are present in the TEI file and indicate the concepts that are described in the textbook's content.

Topics

Topics are shaped via inquiry and discourse, by the way we describe certain phenomena. Given that textbooks are the ultimate resource for these processes, they can serve as an effective input into the modelling of topics. I will now detail some mathematical techniques that can be used to model the abstract structure of topics, some of which are used in our methodology.

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 TF-IDF formula, shown on the screen, has two components. The Term Frequency component measures the importance of a word in a document, while the Inverse Document Frequency penalises terms with high frequency in the corpus, since terms that appear in many documents are less valuable in distinguishing between documents.

Latent Dirichlet Allocation

Latent Dirichlet Allocation is another foundational technique in topic modelling that defines each topic by measuring the likelihood of each word appearing in it. In turn, documents are modelled as mixtures of topics.

\/\/\

Previous research where LDA is used to link documents from different textbooks shows that it is 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, which can significantly influence the model's performance. Also, LDA struggles with documents that have overlapping or tangential topics.

Embedding Models

So, to improve on LDA, embedding models have been implemented to create dense vector representations of words and documents.

\/\/\

Word2Vec, which was published by a team at Google in 2013, is a pivotal example of this approach. It generates vector representations for words in such a way that the spatial relationships between vectors capture semantic relationships between words.

\/\/\

For example, in the analogy “woman is to man what queen is to king”, Word2Vec aims to represent vectors for these words such that the spatial relationship between “woman” and “man” is similar to that between “queen” and “king”, as shown on the screen.

\/\/\

Extending this concept, Doc2Vec allows entire documents to be represented as vectors that consider the overall context and semantic content of the documents. It has been shown that Doc2Vec outperforms LDA in textbook content linkage tasks.

Bidirectional Encoder Representations from Transformers

In 2019, BERT is introduced as a “language representation model” that is pre-trained using two unsupervised tasks. First, a “masked language model” is used, where some input tokens are randomly masked, and the model is trained to predict these masked tokens. Additionally, BERT employs a “next sentence prediction” task, where it learns to predict if a sentence is the subsequent sentence in the original document. This method enables the model to infer the meaning of a sentence more accurately in the case of ambiguity, which is crucial for many downstream 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.

Recurrent Neural Networks

Recurrent Neural Networks are a type of deep neural network that contain self-feedback loops, allowing them to effectively remember sequences in the data.

\/\/\

While RNNs have been shown to perform well, they can face the “vanishing gradient problem”, whereby the gradient of the loss function becomes too small for effective learning, thus making it difficult for the network to learn long-range dependencies.

\/\/\

Long Short-Term Memory networks were introduced to overcome the limitations of traditional RNNs, through the introduction of a structure called a memory cell, capable of maintaining information in memory for long periods. This addresses the vanishing gradient problem and preserves long-term dependencies in sequential data more effectively than traditional RNNs.

Semantic Similarity

Semantic similarity refers to the degree to which two pieces of text are alike in meaning. To determine whether there is any semantic similarity between two pieces of text, they must first be represented as vectors and then compared using cosine similarity.

\/\/\

The classical cosine similarity formula is shown on the screen. Given two non-zero vectors of the same dimension, the cosine similarity is the cosine of the angle between the two vectors. Thus, cosine similarity is unaffected by magnitude – a useful property in text analysis where the lengths of documents can vary greatly.

Linking Multiple Textbooks

Previous research has described two methods for handling the hierarchical structure of the textbook, which can be applied to our vector representations.

The first approach is Topic Aggregation, which only computes the topic vectors for sections at the lowest level of the hierarchy. For higher levels, topic vectors are calculated by taking a weighted average of sub-topic vectors.

\/\/\

The second approach is Re-Indexing, whereby topic vectors are computed for each section by considering a section's content to be that of the section and all its sub-sections.

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.

Methodology

I will now discuss our methodology. We propose a pipeline that fits broadly into two phases.

\/\/\

A prerequisite, however, is textbook knowledge models, which are obtained from previous research in the form of TEI files.

Then, the first phase involves generating a training dataset using unsupervised topic modelling methods. Here, we test a variety of textbook integration methods and use the best one to generate a labelled training dataset.

In the second phase, we obtain BERT embeddings for each section and use these to train neural networks to classify topics within the training dataset.

Dataset Generation

To generate a dataset, we choose a base textbook and integrate the entries from the table of contents of other textbooks, such that the result is a table of contents where each entry points to groups of sections rather than a single section.

Once the textbooks are integrated, we obtain our normalised dataset by setting the class label of each section to be the largest ancestor of the section

in the integrated textbook. As shown on the screen, there are seven highlighted sections across three textbooks. However, these fit into only two class labels: “Overview and Descriptive Statistics” and “Probability”. These are the top-level sections from the base textbook. This approach is taken to ensure that each class has sufficient membership.

\/\/\

For each section, we have wide variety of attributes available to us, to serve as inputs for our integration approaches.

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 used 12 textbooks from the statistics domain when developing our method.

Strategies

We tested several approaches for integrating the textbooks.

First, the TF-IDF method was used to convert each section into a vector. Vectors were compared with cosine similarity and if this value was above a certain threshold, then we were able to consider the two sections a match.

Similarly, we also tried to convert each section to a vector using Doc2Vec.

A slightly different approach was to cluster sections based on their tags, whereby a tag might be the DBpedia concept that the section is annotated with. K-means clustering was used here and sections that are in the same cluster are considered similar.

Based on the common practice of “seeking a second opinion,” it was thought that combining the TF-IDF and Clustering models using an ensemble averaging approach might yield an improved result.

Additionally, a hybrid approach was implemented whereby the TF-IDF and Doc2Vec models were combined iteratively. The TF-IDF model first classifies some matches as ‘definitive’, while maintaining a record of some ‘uncertain’ matches. Simultaneously a Doc2Vec model is fit and applied to matches that

are within the uncertain range for TF-IDF. If Doc2Vec considers the sections a good match, they are added to the resulting integration of sections. The intuition behind this is that the two models have different strengths and weaknesses and combining them might be complementary.

Finally, each of these strategies was tested with an approach whereby, sections were iteratively updated to include the attributes of their newly matched section, and the vectors for each section were recomputed.

Topic Classification

Once the textbooks have been integrated and a new dataset has been generated, then we can begin training an advanced language model for the classification of topics.

\/\/\

The first stage in this process is to obtain vector embeddings for each textbook section using BERT. BERT accepts special tokens, like C.L.S. and S.E.P., 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.

We propose that the additional information allows the model to better learn how concepts are described by content, enhancing its ability to understand and represent the relationship between these two elements in the text.

The text input is then tokenized and passed to a version of DistilBERT from the Hugging Face transformers library, which converts the text into a high-dimensional vector. DistilBERT was chosen as it is shown to be much smaller and faster than the original BERT, while maintaining almost all its language understanding capabilities.

\/\/\

These vectors are then used to train RNNs to classify new sections according to their topic.

Evaluation

Now I'd like to discuss the results that were seen with this proposed methodology.

Dataset Generation

To evaluate the performance of the textbook integration task, we use a manual mapping between two of the textbooks generated by experts. In this context, precision is the proportion of the mappings identified by the algorithm that agree with the experts. 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 wrongly associating two different topics can have a greater cost than failing to associate between two similar ones.
\/\/\
However, this doesn't mean recall is unimportant. Therefore, it was decided to use the F_β score for evaluation, which is the weighted harmonic mean of precision and recall. By setting β to 0.5, we consider recall to be half as important as precision.
\/\/\
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. Out of the individual models, TF-IDF and Doc2Vec were the best performing. Given that the hybrid model improves on results of these models, with particularly good precision, the approach of combining the models 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.

Topic Classification

Three types of datasets were used to test the topic classification process. The first is the dataset generated by the expert's manual mapping. The second dataset is generated by applying the best integration method – that is, the hybrid model – to the same two textbooks that the experts used. The last dataset is generated by applying the hybrid model to all twelve textbooks.

\/\/\

In this phase, a similar approach was taken to select the optimal parameters for the neural network. Additionally, 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.

For each of the optimal models, we compared the performance of the model to that of BERTopic. BERTopic is a topic modelling framework that has several capabilities, including out-of-the-box topic classification.

Given this is a supervised classification problem, we evaluated the performance of each model using accuracy. All the metrics shown in the table were calculated by taking the weighted average across all classes, and the average across 5 folds of cross-validation.

In every metric, our models perform better than the BERTopic baseline. Additionally, across all datasets, our model performs better when concepts are prepended to the input text.

The expert dataset performs better than the generated datasets, and the large generated dataset performs particularly bad.

Limitations

I'd now like to discuss some limitations of our approach.

\/\/\

The poor performance of the large dataset might be due to the large number of classes introduced by integrating all twelve textbooks simultaneously. However, the strong performance of the expert dataset 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.

\/\/\

It should also be noted that performance might be worsened by using the pre-trained Hugging Face DistilBERT implementation. Perhaps with more time or resources, a model that is more specific to our use case could be trained.

\/\/\

The classes in Phase 2 are quite broad, given that top-level textbook sections are used as the class label when generating datasets. Future research might explore a way to obtain more granular classes without degrading performance.

Key Takeaways

These are the key takeaways of my presentation. We commenced with an examination of existing literature, providing a foundation for our subsequent contributions.

\/\/\

Our two-phase methodology, involving dataset generation and topic classification was then introduced, providing the foundations for a novel architecture to develop a domain-dependent and fine-grained topic model.

\/\/\

The subsequent evaluation of this methodology confirmed that while the performance of this architecture is not particularly strong, the initial results do 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.

Thank you.