Concept Extraction From Structured Course Material

Jacob Hoffman
University Of Pittsburgh
jah292@pitt.edu

Raja Krishnaswamy
University Of Pittsburgh
rek94@pitt.edu

Abstract

A key component of enhancing learning management systems is information on the concepts covered in specific course materials, but manually extracting such concepts is a time-consuming and laborious task. To this end, we propose a BERT-based model to automatically extract course concepts, as well as a novel English-language dataset of labeled course materials. The model was fine-tuned using a mix of human-expert labeled and distantly labeled data, which alleviated the heavy workload of manually annotating the new dataset. Results show the addition of distant labels improves the performance of the model (with 10% F1-score improvement.)

1 Introduction

Course Concept Extraction is the task of extracting meaningful key terms called concepts from a given course. This task can be used in learning management systems to expedite the learning process for students and help students better understand the main points of the material. However, manually labeling such concepts is a time-consuming and laborious task. If such a system is to be used in a university, professors won't have the time to necessarily do it by themselves. Thus, there is a need to automate the process. Due to the abundance of materials such as slides and syllabi, extracting course concepts from them would be useful in a university setting. However, there are no known English datasets for this task and only two datasets in Chinese that are related to this task, which are the dataset used in DS-MOCE (Lu et al., 2023) and MOOCCube (Yu et al., 2020).

The paper contributes a BIO-labeled English dataset consisting of concept-labeled slides from 6 courses from the University of Pittsburgh's Computer Science Department. Additionally, a Course Concept Extraction model was trained on the created dataset. Due to the excessive labor required to manually annotate a course, distant labeling was used to generate more training data. Results show that including data generated using distant labeling improves the f1-score by 10%.

2 Prior Work

To illuminate the field of study, we shall examine some prior papers in the field. We focus here on papers without labeled data, as we do not have it either.

The first works in the field of course concept extraction used statistical methods to extract concepts in an unsupervised way. Some works in this vein are TextRank, Embedding-Based Graph Propagation, and SemKeyphrase, all of which are described below.

TextRank's approach was to represent text as an undirected graph of words based on which words collocated with other words in a certain window (Mihalcea and Tarau, 2004). They then perform a variation of the page rank algorithm used in the Google Search Engine to assign scores to the words, of which, the ones who exceed a certain threshold are considered concepts. This algorithm initially starts off each node with random scores, and then based on the average of the scores of the nodes connected to it, each node gets an updated score. This continues until the scores change by less than a predetermined threshold. Finally, they combine adjacent concepts into a single concept. To test their model, they ran their algorithm on 500 abstracts from the Inspec database, which contain associated keywords as gold labels, and determined the precision, recall, and f1-score of the model given various parameters on the test set. Using these, they found that the algorithm outperformed the state of the art supervised model at the time.

A little more recently, however, the problem of low-frequency keyphrases has shown up when performing such algorithms on MOOCs, which

have been attempted to be addressed by unsupervised learning models like Embedding-Based Graph Propagation by Pan et al. in 2017 (Pan et al., 2017). This model learns the latent word embeddings of candidate concepts from online encyclopedias and then ranks the concepts using a graph-based propagation algorithm. This algorithm is similar to TextRank, but it has an initial seed-set of known concepts whose scores are initialized as one and modifies the scoring function to introduce a penalty for concepts that share words among other modifications. They then evaluated the model using different courses from both XuetangX and Coursera using the R-precision and Mean Average Precision metric and found that it outperforms all other methods.

In "A novel cluster-based approach for keyphrase extraction from MOOC video lectures" (Albahr et al., 2021), the automatic extraction of keyphrases from MOOC video lectures is attempted using SemKeyphrase, which is an unsupervised cluster-based approach that incorporates a new semantic relatedness metric and a two-phase ranking algorithm called PhaseRank. The first phase is a ranking of candidates, and the second phase is a re-ranking of the top-candidate keyphrases. Since keyphrase extraction from MOOC video lectures suffers from low-frequency and delayed occurrences of important keyphrases, SemKeyphrase utilizes a semantic relatedness metric that enables leveraging the semantic relations inferred from the ranking candidates' word embedding vectors and co-occurrence relation between the candidate keyphrases and the contextual candidate keyphrases in the MOOC video lectures. The SemKeyphrase approach outperformed three common baselines and the state-of-the-art approach at the time, including an improvement on the TextRank model, PositionRank. Finally, the authors describe how they built their own MOOC dataset based on video lectures collected from Coursera. Then, the approach in Pan et al. is used to manually create a gold-labeled keyphrase dataset for evaluation purposes, which we believe will be a useful approach for our project.

These unsupervised models have been the most prevalent in concept extraction with minimal data. However new approaches for extending small amounts of data for the purposes of training have been attempted.

In the "Leveraging Book Indexes For Automatic Extraction Of Concepts In MOOCs", the authors develop an approach for automatically extracting the major concepts from a MOOC (Boughoula et al., 2020). First it labeled the entire textbook with concepts from the index at the back of the book. Then, they trained a neural network, using the labeled textbooks as training data for a supervised machine learning algorithm instead of manually-labeled annotated data. This is an interesting approach to the lack of data, but isn't that great. In the paper they trained three models on three different textbooks and applied them to two MOOC courses in the area. They found that the model's performance depended heavily on the textbook training data and how well it covered the topics in the MOOC. They found that the model was sensitive to the way the concepts were represented, causing false false-positives and negatives.

The idea of labeling incomplete data automatically for training purposes has not been limited to that paper. Recently, the paper by Lu et al, a three-stage framework called DS-MOCE was developed in an attempt to reduce extreme noise and incomplete annotation due to a limited dictionary of concepts and diverse MOOCs (Lu et al., 2023). DS-MOCE first leverages pre-trained language models for dictionary empowerment, i.e. labeling each concept in the dictionary with a discipline, to reduce this noise. Then it uses this to do Distant Supervision Refinement, where it generates labels for the incompletely annotated content by labeling the incomplete data with concepts from the dictionary that are most likely to be used in the data's discipline. This solves the incomplete data problem. Finally, they produce discipline-embedding models trained with a self-training strategy based on label-matching for refining concept extraction across academic disciplines, which is done to further reduce noise. The paper explores two self-training strategies, the two student-teacher networks co-training strategy and the positive-unlabeled learning loss strategy.

The paper also provides an expert-labeled dataset that covers 20 academic disciplines. Although this dataset is in Chinese, it provides guidance for the English dataset we plan to create. They evaluate this model using precision, recall, and F1-score and compare it to other distantly supervised models and one supervised one. Both the models outperformed the state-of-the-art distantly supervised methods, but was still dwarfed by the FLAT supervised learning model. The co-training model did very well in the precision metric (81.93), even scoring high above the state of the art Supervised

learning model (56.03), but had a slightly less impressive Recall score (30.82), which was below the state-of-the art distant supervision model (BOND) (44.78). Now, the Positive Unlabeled Learning strategy was able to get the highest recall score of a distantly supervised model (49.34), but was not the best when it came to precision, scoring 34.53. Both models received high F1-scores, the co-training strategy getting 44.79 and the Positive Unlabeled Learning scoring 40.62, better than the other unsupervised models.

Interestingly, none of these papers have examined course slides and syllabi, instead opting to examine lecture transcripts. We aim to do this in our model.

3 Text Extraction and Pre-processing

To begin annotation, we have created a Python script to process course slide pdf files into a format that can be annotated. First, we use the Python library pdf2text to extract the lines of text from the pdf files in the course slide dataset, and then a separator is added between slides to mark the break. The next step is to tokenize the lines of text using the nltk tokenizer. The tokenized lines of text are output into json files. These output files contain the structure of the unlabeled dataset, which consists of pairs of tokenized text and BIO labels for each token, i.e. a label for each word determining if it is the beginning of a concept, inside a concept, or not a concept at all. A labeled entry in this dataset is in the following form:

```
{
“text:” [“The”, “Operating”, “System”,
    “uses”, “Interrupts”, “to”,
    “implement”, “System”, “Calls”],
“label”: [”O”, ”B”, ”I”,
    ”O”, ”B”, ”O”,
    ”O”, ”B”, ”I”]
}
```

4 Dataset

For the new dataset, the project team manually annotated the lines extracted from the course slide pdf files of 2 courses (CS-0441 and CS-1567), and there are 2 more in progress (CS-0449 and CS-1541). These courses are from those taught in the CS Department of the University of Pittsburgh over the past 4 years. The manually labeled data amounted to 10,659 lines of the 46,195 total lines contained in the 6 courses that were selected for annotation (which are CS-0441, CS-0449, CS-1541, CS-1550, CS-1567, and CS-1622), which is 23.1% of the dataset.

During the manual annotation process, a word/series of words had been defined as a concept based on whether it was a defined term and whether it was used repeatedly (more than 2 times), or extensively focused on.

On average, it has taken 30 minutes to annotate a 30-page slide deck and around 10 minutes for a 15-page slide deck.

The manually labeled dataset was split 80:20 between training and testing data for training and evaluating the concept extraction model.

5 Distant Supervision

A dictionary with 1,137 concepts was compiled from multiple Wikipedia articles related to the “Computer Science” field: "List of Terms Relating to Algorithms and Data Structures" (Wikipedia contributors, 2023d), "List of Computer Term Etymologies" (Wikipedia contributors, 2023c), "Glossary of Computer Science" (Wikipedia contributors, 2023a), "Index of computing articles" (Wikipedia contributors, 2023b), and "Compiler Construction/Glossary" (Wikibooks, 2018).

The project team implemented a simple form of dictionary empowerment, which was used to automatically label the remaining course slide data. The routine involved iterating through all of the words, and if any of the concepts in the dictionary matched with the successors of the current word, the concept with the most matching tokens was selected as a label for the concept.

The distant labeled dataset was appended to the manually labeled training data for training the concept extraction model.

6 Model

The project team fine-tuned the bert-base-uncased model included in the Hugging Face Transformers library along the line of NielsRogge’s tutorial (NielsRogge). This will be used as a general baseline for the concept extraction task using the labeled dataset. In conjunction with the distantly labeled data, 80% of the manually labeled data was randomly sampled to train the model.

Includes Distant Labels	Accuracy	Precision	Recall	F1-Score
No	0.92	0.48	0.42	0.45
Yes	0.93	0.50	0.62	0.55

Table 1: Evaluation metrics using the bert-base-uncased model.

7 Results

Using 20% of the manually labeled data that was set aside earlier, the trained concept extraction model was evaluated on a model trained using distant labeled data and another model that did not use distant labeled data during training (refer to Table 1).

The usage of distant labeling improved the F1-score by 10%. The project team believes this indicates that distant labeling is useful for improving the results of the concept extraction model.

8 Ethics

One potential ethical issue is that our dataset is based on academic course materials, which have to be sourced from professors' intellectual property. Thus, one issue is how we make sure that using their data in our model is acceptable, while also having enough data to train the model. Asking for permission is a simple solution; however, if some person is unreachable, their course material may be discarded. Another potential ethical issue is that a student might apply the model to a section of text they were assigned with the hope of getting the main topics so that they may circumvent the course work, which would not result in an ideal learning result. This is one of many possible outcomes where the tool could be potentially used in a way that will ultimately leave the student with a shallower understanding of the topics.

9 Individual Contributions

Both Raja Krishnaswamy and Jacob Hoffman did a substantial amount of work for the project.

Raja Krishnaswamy annotated the CS-0441 course slides and a portion of the CS-0449 slides (70% of the manually annotated data), compiled 25% of the concept dictionary, implemented the distant labeling algorithm, and adapted the model code written by Jacob to work with the distantly labeled data.

Jacob Hoffman annotated the CS-1567 slides and a portion of the CS-1541 slides (30% of the manually annotated data), compiled 75% of the concept dictionary, and implemented the BERT-based concept extraction (BIO-labeling) model.

10 Future Work

The project team believes there are many options for improving the performance of the model.

The dataset should be dramatically increased in size. The dataset currently includes only course material including computer science concepts, but ideally there should be course material spanning all of the major education paradigms. Additionally, the team should also include syllabi from the courses in the dataset. Furthermore, the concept dictionary should be expanded in this regard and should include labels of what paradigm each concept is associated with. This will help the project team to improve the granularity of the distant labeling utility. The project team currently has collected the course pdf slides for 13 courses at the University of Pittsburgh, but only 6 of these were considered for this paper.

Although there are initiatives for optimizing the amount of data that needs to be manually annotated, the project team still does require more data to be professionally labeled by hand. This will have to be done by the team, or some form of crowd-sourcing fulfilled by university members.

The project team plans to explore an implementation of self-learning with the model through the use of teacher-student self-learning process.

Although a model was trained and evaluated on the project dataset, there was not much effort in evaluating different models, hyper-parameters, or frameworks, so we plan to evaluate them in the future.

Acknowledgements

Thank you to Dr. Yoder and Pantho for being incredibly helpful with the project.

References

Abdulaziz Albahr, Dunren Che, and Marwan Albahar. 2021. A novel cluster-based approach for keyphrase

extraction from mooc video lectures. *Knowledge and Information Systems*, 63(7):1663–1686.

Assma Boughoula, Aidan San, and ChengXiang Zhai. 2020. Leveraging book indexes for automatic extraction of concepts in moocs. In *Proceedings of the Seventh ACM Conference on Learning @ Scale*, L@S '20. ACM.

Mengying Lu, Yuquan Wang, Jifan Yu, Yexing Du, Lei Hou, and Juanzi Li. 2023. Distantly supervised course concept extraction in moocs with academic discipline. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

NielsRogge. Fine-tuning bert for named-entity recognition. https://github.com/NielsRogge/Transformers-Tutorials/blob/master/BERT/Custom_Named_Entity_Recognition_with_BERT.ipynb. Accessed: 2023-12-14.

Liangming Pan, Xiaochen Wang, Chengjiang Li, Juanzi Li, and Jie Tang. 2017. Course concept extraction in MOOCs via embedding-based graph propagation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 875–884, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Wikibooks. 2018. Compiler construction/glossary — wikibooks, the free textbook project. [Online; accessed 14-December-2023].

Wikipedia contributors. 2023a. Glossary of computer science — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Glossary_of_computer_science&oldid=1180721185. [Online; accessed 16-November-2023].

Wikipedia contributors. 2023b. Index of computing articles — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Index_of_computing_articles&oldid=1173520671. [Online; accessed 16-November-2023].

Wikipedia contributors. 2023c. List of computer term etymologies — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=List_of_computer_term_etymologies&oldid=1177332180. [Online; accessed 16-November-2023].

Wikipedia contributors. 2023d. List of terms relating to algorithms and data structures — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=List_of_terms_relating_to_algorithms_and_data_structures&oldid=1181598111. [Online; accessed 16-November-2023].

Jifan Yu, Gan Luo, Tong Xiao, Qingyang Zhong, Yuquan Wang, wenzheng feng, Junyi Luo, Chenyu Wang, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jie Tang. 2020. Mooccube: A large-scale data repository for nlp applications in moocs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.