Project LexiMap: Automated Extraction and Mapping of Textbook Keywords

Kennesaw State University

CS 7267: Machine Learning

Dr. Jiho Noh

Clark Pfohl

bpfohl1@students.kennesaw.edu

Ian Taylor itaylo19@students.kennesaw.edu

Michael Zimmerman

mzimmema@students.kennesaw.edu

***Abstract*—This project investigates the effectiveness of automated keyword extraction using KeyBERT, a BERT-based [1] keyword extraction tool, by comparing machine-extracted keywords against the textbook’s index. We evaluate the performance across different chapters using precision, recall, and F1 score metrics implemented with fuzzy matching to account for variations in keyword representation. Our initial results show promise in automating the keyword extraction process, and we plan to map the machine-extracted keywords to a knowledge graph.**

1. INTRODUCTION

Keyword extraction is a fundamental task in natural language processing (NLP) that involves identifying the most important words or phrases that represent the main topics of a document. Recent approaches leverage transformer-based models like BERT to capture semantic meaning.

In this study, we evaluate KeyBERT, which uses BERT embeddings to extract keywords and phrases from documents. Our research question is: How well can automated keyword extraction using KeyBERT match human-curated keywords in educational texts? We use a textbook, *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow,* and its index as our dataset, providing a real-world case study for educational content indexing.

1. METHODOLOGY

*A. Dataset*

The dataset is derived from the textbook [2] containing a total of 19 chapters. Keywords are extracted and organized by chapter. The human-curated index is used as the ground truth for evaluation purposes. By using this approach, we have a standard of comparison to evaluate our keyword extraction dataset. Chapters 1-5, 7-9, 13-19 will be used to train, and chapters 6, 10, 11, and 12 will be used as the test set.

*B. Keyword Extraction*

We implemented a keyword extraction system using the following components: KeyBERT with a transformer model (the sentence transformer model `all-mpnet-base-v2`) [3], which provides strong semantic understanding while being computationally efficient; an n-gram range of 1-3 words to capture both single terms and phrases, a diversity parameter of 0.7 to ensure variety in extracted keywords, and a `maxsum` algorithm to generate optimal keyword diversity.

*C. Evaluation Metrics*

The system’s performance was evaluated with Fuzzy Matching to handle the variations in keywords between the human-generated and machine-extracted keywords. This was achieved through text normalization (lowercase and special character removal), exact match detection, substring containment evaluation, and sequence matching for fuzzy comparison.

*D. Optimization*

We implemented optimizations after running into performance issues. First, a caching system was implemented to maintain consistent information across the training and the evaluating stages. This shared cache allows for significant speed improvements for repeated runs. Next, error handling was introduced to log the stages of the output; this allows for robust error recovery and result saving to prevent data loss.

*E. Knowledge Graphs*

At this stage in our project, the knowledge graphs have not been created yet. The conceptual model is present-- using NetworkX, take the extracted keywords and utilize the semantic understanding from KeyBERT to create edges between them.

1. INITIAL RESULTS

Our preliminary results show that KeyBERT can successfully extract relevant keywords from textbook chapters, but with varying degrees of success depending

* 1. *Performance Metrics*

We extracted keywords from the combined training chapters (1-5, 7-9, 13-19) and used these as a baseline for comparison. The average precision at this stage is 0.059 (only 5.9% of extracted keywords match ground truth); the average recall is 0.044 (only 4.4% of ground truth keywords are found); the average F1 score is 0.049 (harmonic mean is very low). On average, fewer than 3 correct keywords were found per chapter, most extracted keywords don’t match ground truth, and most ground truth keywords aren’t found.

* 1. *Issues Identified*

Thus far, we have struggled capturing the correct keywords-- the snags have been identified as following: partial matching issues cause too many terms to be captured and overly generic keywords are not filtered out.

* 1. *Changes Implemented*

Initially, we found that the model we were using was underperforming; we switched from “all-MiniLM-L6-v2” to “all-mpnet-base-v2”. The latter model does slightly better. Additionally, after implementing hierarchical matching and N-gram matching, we have seen a 273% increase in average precision, a 66% increase in average recall, and a 124% increase in average F1 score.

1. CHALLENGES AND SOLUTIONS
   1. Set Up

With the Python environment and the many dependencies needed to run the ML elements, there were difficulties running the initial code. The solution came by using Google-Colab and virtual machines.

* 1. Collaborating

Our team needed a reliable way to share code and data while tracking our changes. We established a Git repository and used GitHub for collaboration and version control. Regular virtual meetings helped coordinate efforts and ensure alignment on project goals.

* 1. Data Extraction

Much of the heavy lifting of extracting text from the chapters of the textbook had been done by our professor, formatting each chapter as a .txt file. The challenge then became parsing through the individual chapter files-- using UTF-8 encoding solved this problem.

* 1. Evaluating the data

1. NEXT STEPS
   1. Compiling results

We have gotten the keywords out of the data set but we have not yet arranged them to be of use to the mind map. In the coming weeks we will need to manipulate the results to be in a better format for our next step.

* 1. Mind Map

After we complete data results compiling we are going to write a program to connect them with a map of the words. Each word with connections to others data points.

* 1. Testing the results

Our team is going to compare the data from the mind map to the index we originated the dataset from to see if we are accurate with our connections. Should we fail at this step we are going to go back to extraction and review all steps until we find the issue.

1. TEAM CONTRIBUTIONS
   1. Clark

* Established Git repository
  1. Ian
* Developed KeyBERT training model
  1. Michael
* Led documentation and report writing

1. REFERENCES

[1] M. Grootendorst, ‘KeyBERT: Minimal keyword extraction with BERT’. Zenodo, 2020.

[2] A. Géron, “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems”.

[3] “Pretrained Models — Sentence Transformers documentation,” www.sbert.net. https://www.sbert.net/docs/sentence\_transformer/pretrained\_models.html