Evaluation and Modification of Local Citation Recommendation Systems: A Comparative Study of Available Embedding and Similarity Search Techniques.

Kabir Jaiswal, Nigam Patel, Spriha Jha

New York University
kj2294@nyu.edu, np2726@nyu.edu, sj3520@nyu.edu
GitHub Repository: sprihajha/dl-final-project

Abstract

This study explores the enhancement of the two-stage Local Citation Recommendation System, initially studied in work 'Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-based Re-ranking.' We have tried to create an alternative system using OpenAI embeddings and OpenAI and Cohere for re-ranking. We conduct an in-depth comparative analysis of these different models, focusing on their impact on the effectiveness of the citation suggestion pipeline. Despite resource constraints limiting us to 100 re-ranked candidates, our findings provide significant insights into the potential advantages and drawbacks of various embedding and similarity search methods. Our new method managed to achieve approximately 5% better accuracy than the one presented in the paper.

Introduction

Citation recommendation systems are crucial in academic research, enabling researchers to find relevant literature and acknowledge previous work efficiently. However, the rapid growth of academic literature presents an increasing challenge for these systems. It becomes difficult to efficiently and accurately recommend citations from an exponentially growing pool of candidates. In the face of this challenge, the paper' Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-based Re-ranking' [2] proposed a two-stage Local Citation Recommendation System that achieved promising results.

In this paper, we aim to build upon this work, analyzing the efficacy of the original system and exploring potential improvements. We propose two implementations utilizing OpenAI embeddings, re-ranking with the LLMChain Extractor, and Cohere Re-rank. Through this comparative study, we aim to gain insight into how different embeddings and similarity search methods affect the effectiveness of the citation suggestion pipeline. To do this comparative study we use the arXiv-200 dataset proposed in the original paper.

Literature Survey

The literature survey of this study primarily focuses on the following:

- *Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-based Re-ranking Paper:* If available, this work recommends a reference from the local citation, and a global context needs to be added. The researchers proposed a hierarchical Attention encoder (HAtten), which, when coupled with a SciBERT [3] re-ranked fine-tuned on local citation recommendation tasks, achieves high prefetch recall for a given number of candidates.

- *LangChain:* This is a framework for developing applications powered by language models. It connects a language model to other data sources and allows the language model to interact with its environment. It assists in developing applications such as question-answering over specific documents, chatbots, and agents. It provides out-of-the-box support to build NLP applications using Large Language Models (LLMs) and connect to various data and computation sources [5].

- *OpenAI Embeddings:* OpenAI embeddings are numerical representations of concepts converted to number sequences, making it easy for computers to understand their relationships. OpenAI's text embeddings measure the relatedness of text strings and are used to personalize, recommend, and search content. The normalized embeddings are of length 1, which makes cosine similarity computation faster. These embeddings outperform top models in three standard benchmarks, including a 20% relative improvement in code search [6].

- *Pinecone*: Pinecone is a vector database designed to handle all the complexities and algorithmic decisions behind the scenes. It is a managed, cloud-native vector database with a simple API and no infrastructure hassles. The database is designed for high-performance vector search applications and offers ultra-low query latency at any scale, even with billions of items [8].

- *Cohere AI*: Cohere is a platform that enables developers to build robust applications with large language models. It uses a transformer-based architecture and improves human-machine interactions [7].

Methodology

Model Training and Testing

Our methodology is centered around a comparative study of the Local Citation Recommendation System as researched in the paper titled 'Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-based Re-ranking,' and two distinct implementations that utilize OpenAI embeddings with different re-ranking methods. These implementations involved using LangChain, OpenAI, Cohere, and Pinecone tools.

We constructed our code based on the authors' provided GitHub code[4] for the abovementioned paper. The code served as our foundation, which we then modified and extended to fulfill the requirements of our study.

In both our implementations, the crux used OpenAI embeddings to encode the text data and stored it in a PineCone vector store. OpenAI embeddings are high-dimensional numerical representations of textual data that capture the semantic content of the text. Each piece of text, be it a word, sentence, or paragraph, is converted into a vector in a multi-dimensional space where the relative positions of the vectors reflect the semantic relationships between the texts they represent.

The essence of these embeddings is that texts with similar meanings represent vectors close to each other in the vector space. In contrast, texts with distinct meanings represent vectors that are further apart. This characteristic is essential for tasks such as recommendation, search, and personalization, where measuring the semantic similarity between pieces of text is a common requirement. To get the initial list of documents for re-ranking we would perform a cosine similarity search on the vector store.

The re-ranking methods used in each implementation were as follows:

- *LLMChain Extractor:* In the first implementation, we used the LLMChainExtractor in the LangChain library to develop a re-ranking method. This method enables us to use the GPT-3.5-turbo model for re-ranking.
- *Cohere Re-rank:* The second implementation employed the re-ranking solution provided by Cohere AI. Cohere AI is a platform that allows developers to leverage natural language understanding capabilities in their applications.

We performed a series of tests and evaluations on a defined dataset to evaluate the effectiveness of these re-ranking methods. In addition, we tuned parameters for each re-ranking method to achieve optimal performance in speed and accuracy. The results were then compared against each other and the results obtained from the original Local Citation Recommendation System.

The dataset used for this evaluation was defined in the original paper and is known as arXiv-200[9]. This is a dataset for local citation recommendation, consisting of 3.2 million local citation sentences along with the title and the abstract of both the citing and the cited papers. Around 1.66 million papers' titles and abstracts are available in the database.

This methodology allowed us to perform a thorough comparative analysis of the effect of different embeddings and similarity search methods on the citation suggestion pipeline. The results of this study will provide valuable insights into the strengths and limitations of each method, guiding future work in this area.

Results

In this section, we present the results of our comparative study of the three different implementations of the citation recommendation system 1 2. These results are presented in tables, plots, and other supplementary images to illustrate each system's performance and effectiveness clearly.

Due to resource constraints we only considered the first 100 candidates for re-ranking and scored it based on the first 100 queries in the test data set. The results can be see in table 1.

We can see from these results that the OpenAI embedding with Cohere Re-ranking performed the worst followed by the original method proposed in the paper which is Hierarchical-Attention Text Encoder and SciBERT-based Re-ranking. The OpenAI embedding and LLMChainExtractor Re-ranking which uses OpenAI's GPT-3.5-turbo model performed the best in our testing.

Moreover the OpenAI embedding and LLMChainExtractor Re-ranking which uses OpenAI's GPT-3.5-turbo model uses considerably less resources on the end users machine as we are using a pre-trained model on OpenAI's servers instead of having to train a model on our own machine. This also improves compatibility as it can now run on machines which don't have CUDA support or capabilities.

Our results provide a comprehensive analysis of how different embeddings and similarity search methods can impact the effectiveness of a citation recommendation system. These findings will serve as a valuable resource for future research.

Limitations:

While our study provides a comprehensive comparison between different implementations of a citation recommendation system, it has limitations. The following are some of the notable limitations of our study:

- *The number of Re-ranked Candidates:* We only considered a maximum of 100 candidates for re-ranking in our implementations due to resource constraints. The resource constraint limited the scope of our analysis and

Method	R@10 for 100 re-ranked
HATE and SciBERT Re-ranking	0.4500
OpenAI embedding and Cohere Re-ranking	0.1100
OpenAI embedding and LLMChainExtractor Re-ranking	0.4777

Table 1: Results of different encoding and re-rankings on the arXiv-200 dataset

may only partially represent the performance of the systems in scenarios with more candidates.

- *Dependence on External Tools and Libraries:* Our implementations rely on external tools and libraries such as LangChain, Cohere, and Pinecone. While these tools offer potent capabilities, they also introduce dependencies and potential sources of errors outside our control. Any changes or issues with these tools could impact the performance and reliability of our systems.
- *Limited Evaluation Metrics:* Our system evaluation uses only the accuracy of the recommendations as the evaluation metric. Our evaluation did not consider other factors, such as system speed, user experience, or robustness.
- A *Dataset Limitations:* The dataset used for our study may only partially represent the diversity of real-world citation contexts, thus limiting our results' generalizability.
- A *Fixed Embeddings:* We utilized OpenAI embeddings in all our implementations. While these embeddings perform well in various tasks, they may only be optimal for some citation recommendation scenarios. Our systems' performance could be improved by exploring different types of embeddings or custom-tailored embeddings for citation recommendation.
- A *Scalability and Performance:* While our study provides insights into the effectiveness of different similarity search methods, we needed to thoroughly evaluate the scalability and performance of these methods in a real-world, large-scale environment. Factors such as query latency, computational cost, and storage requirements could significantly impact the feasibility of these methods in practice.

Despite these limitations, our study provides valuable insights into using embeddings and similarity search methods in citation recommendation systems. These limitations also present opportunities for future work in this area.

Conclusion

Our original intent was to conduct a comparative analysis of different embeddings and similarity search techniques, specifically GloVe, OpenAI, and Cohere embeddings, along with cosine similarity, Pinecone vector search, and Nomic's Atlas similarity search methods in the context of the two-stage local Citation Recommendation System (CRS) as proposed by Gu et al. In addition, we aimed to examine the prefetching stage of this CRS pipeline and analyze the impact of different embeddings and similarity search methods on its performance.

The primary focus of our study included an in-depth literature survey of Citation Recommendation Systems, text embeddings, and similarity search techniques. We also planned to use datasets suggested by Gu et al. and the Hierarchical-Attention Text Encoder for the prefetching stage, followed by a SciBERT-based re-ranker for the re-ranking stage.

During our work, however, we found it more practical to use OpenAI embeddings across all our implementations due to their demonstrated performance in capturing semantic information and the ease of integration with our systems. This decision deviated from our original plan of comparing GloVe, OpenAI, and Cohere embeddings. We also made adjustments to our similarity search methods. Instead of Nomic's Atlas, we used Pinecone's cosine vector search and two different re-ranking methods: the LangChain LLMChain Extractor using OpenAI's GPT-3.5-turbo model and the Cohere Re-rank. This change was driven by simplifying our methodology and focusing on tools readily compatible with our chosen embeddings and CRS pipeline.

Despite these changes, we were able to thoroughly analyze the effects of OpenAI embeddings in combination with different re-ranking methods on the performance of the CRS pipeline. Our results provide a comprehensive comparison and valuable insights into the performance of these methods in the context of citation recommendation systems. Our work highlights the potential of OpenAI embeddings and advanced re-ranking methods in enhancing the performance of citation recommendation systems. However, it also underscores the need for further research and experimentation with different types of embeddings and similarity search methods to optimize these systems. The limitations identified in our study, such as the number of re-ranked candidates and dependencies on external libraries, present opportunities for future work in this area.

In conclusion, although our implementation did not strictly adhere to the original plan, adjustments were necessary to ensure the practicality and relevance of our study. Nevertheless, our findings contribute to the broader understanding of how different embeddings and re-ranking methods can be effectively utilized in citation recommendation systems and pave the way for future research in this field.

References

[1] OpenAI, *ChatGPT: An Advanced Conversational AI Model based on GPT-4*, OpenAI, 2021, https://www.

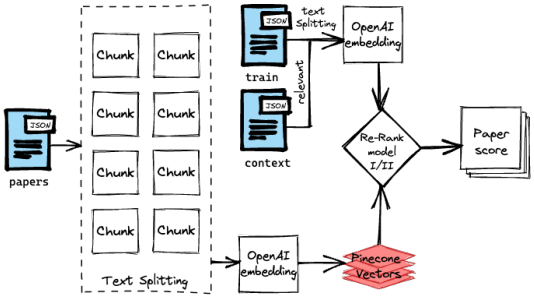

Figure 1: Model Architecture

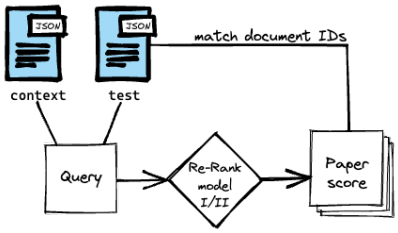

Figure 2: Model Validation

openai.com/chatgpt.

[2] N. Gu, Y. Gao, R. H.R. Hahnloser, "Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-based Reranking", 2022, [Online]. Available: https://arxiv.org/abs/2112.01206

[3] Iz Beltagy, Kyle Lo, Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, November 2019. Association for Computational Linguistics, Hong Kong, China. URL https://aclanthology.org/D19-1371. DOI 10.18653/v1/D19-1371.

[4] *GitHub* Local-Citation-Recommendation https://github.com/nianlonggu/Local-Citation-Recommendation

[5] *LangChain*. Harrison Chase. https://python.langchain.com/

[6] *Embeddings - OpenAI API*.https://platform.openai.com/docs/guides/embeddings

[7] *Cohere AI*. https://www.cohere.ai/

[8] *Pinecone*. https://docs.pinecone.io/docs/overview

[9] *arXiv-200*. https://paperswithcode.com/dataset/arxiv-200