FRI Natural language processing course 2025
Project report

University *of Ljubljana*
Faculty *of Computer and Information Science*

'; DROP TABLE TEAMS; –

Henri Sellis, Igor Sitek

Abstract

For our Natural Language Processing course group project, we are developing a conversational agent that retrieves additional information from Google Scholar documents, to increase the quality of answering questions, ensuring up-to-date outputs. To accomplish this, first a number of most relevant tags are extracted from the user input, then corresponding queries are made to retrieve documents from Google Scholar, and finally the documents, alongside with the original user input, are passed to a Large Language Model, to generate a paraphrased response.

Keywords
Retrieval-Augmented Generation, RAG, web scraping, rank algorithms, Python, keyword extraction

Advisors: Aleš Žagar

Introduction

Large Language Models have proven to be very successful at general-knowledge topics, but a trained model cannot continuously stay up to date with the most recent academic topics. To solve this problem, we are using Retrieval-Augmented Generation to provide an LLM with the most relevant academic papers to allow the LLM to generate more relevant and accurate responses. The documents are retrieved from Google Scholar, as it is an easily accessible repository with lots of academic papers.

With the increasing number of scientific papers published across the world, it is difficult to analyze the most relevant data. Automated literature lookup based on relevant datasets might highly improve the process of conducting scientific experiments, mitigating the risk of work duplication. However, it is crucial to ensure up-to-date answers, thus relying on static datasets is not the most optimal approach. RAG-augmented query analyses alongside with web crawling and web scraping techniques seem to be promising to resolve that matter.

Methodology

The proposed methodology scheme is presented on figure 1. Each vital step is descibed in the following subsections of this article.

Keyword extraction

In order to be able to provide relevant literature, our approach focuses on keyword extraction both from scientific papers and user input. This enables us to create a lookup in available works that would be further parsed.

To further specify our domain, we will limit this research to English papers only, eliminating the necessity of language-independent vector transitions and specific rules in different languages. The English was chosen as it is the most popular language with the highest amount of data available, and it is characterized by low sentence sensitivity (in opposition to romance languages). That characteristic enables low-effort word stemming in text processing [1].

Work [2] is a review of existing methods for keyword extraction, containing summaries of different supervised and unsupervised algorithms. Upon these, Neural Networks seem to be used mainly. On the other hand, ready-to-use implementations of RAKE and RANK algorithms (like TextRank [3] - https://github.com/davidadamojr/TextRank or TopicRank [4] - https://github.com/Aayushpatel007/topicrankpy) seem to be useful enough for our case. Further training using these algorithms will be conducted with relevant datasets.

Model evaluation and learning rate will be measured by the statistical indicators as proposed in work [5]: these include TF-IDF, co-occurrence, and similarity-based values.

Corpus analysis

Following datasets were found in order to fine-tune existing algorithms:

- https://github.com/SDuari/Keyword-Extraction-Datasets,
- https://github.com/LIAAD/KeywordExtractor-Datasets,

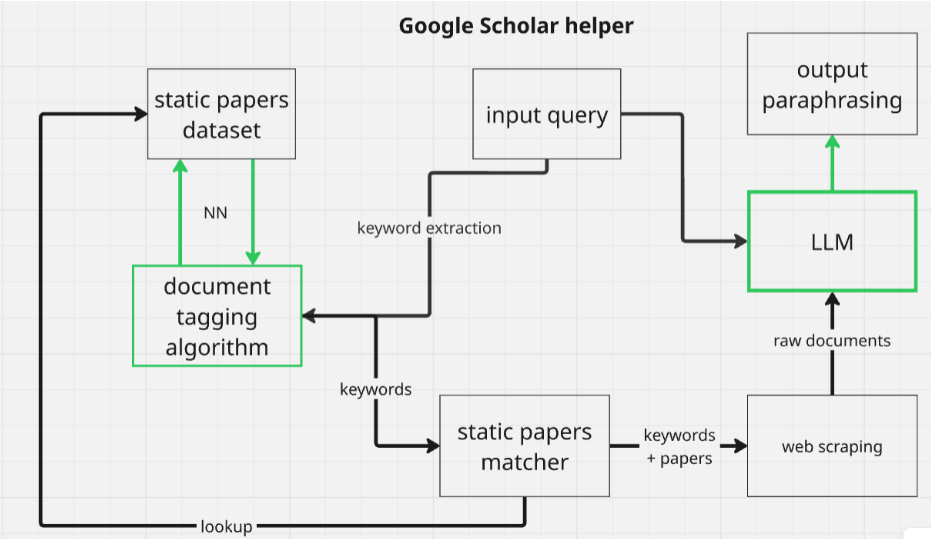

Figure 1. System scheme High-level overview of chat-bot system proposed in this paper.

- https://github.com/boudinfl/ake-datasets,

All of them consist of zip folders with two sub-folders, one containing whole papers in TXT format, and one with manually gathered keywords and key-phrases. These files are matched based on a common key - included in the filename.

Keyword lookup

After parsing user input and resolving its keywords, the next step consists of static dataset lookup in order to find relevant papers. The algorithm is yet to be specified - whereas Levenshtein distance [6] might be useful, it will probably not be enough since multiple subsets of keywords will be searched for.

Web scraping

Provided keywords and user input are combined into a suitable Google Scholar query, in order to assure up-to-date answers. Script is going to be implemented in Python language, using one of the open-source web-scraping packages.
Having in mind commonly used firewalls on publisher websites, some tools like Sci-Hub are considered to bypass them and streamline the process of data scraping. Knowledge must be free!

Data combining and paraphrasing

All the relevant documents (found in the dataset and downloaded from the web) are going to be combined with user input and passed into a pre-trained LLM model. Multiple open-source models will be tested and manually validated. Fine-tuning the LLM model and prompt engineering techniques should provide a satisfactory and reliable output to the user.

References

[1] Peter D Turney. Learning algorithms for keyphrase extraction. *Information retrieval*, 2:303–336, 2000.

[2] Sifatullah Siddiqi and Aditi Sharan. Keyword and keyphrase extraction techniques: a literature review. *International Journal of Computer Applications*, 109(2), 2015.

[3] Suhan Pan, Zhiqiang Li, and Juan Dai. An improved textrank keywords extraction algorithm. In *Proceedings of the ACM Turing Celebration Conference - China*, ACM TURC '19, New York, NY, USA, 2019. Association for Computing Machinery.

[4] Adrien Bougouin, Florian Boudin, and Béatrice Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International joint conference on natural language processing (IJCNLP)*, pages 543–551, 2013.

[5] Nazanin Firoozeh, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. Keyword extraction: Issues and methods. *Natural Language Engineering*, 26(3):259–291, 2020.

[6] Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.