An Intelligent Search Engine Module for Academic Networks

Md Mahbubur Rahman (拉曼) – 3820231103

Department of Computer Science and Technology, Beijing Institute of Technology.

Abstract: *In the realm of academic networks, several key challenges plague the efficiency of information retrieval systems. The current academic search engines often fall short in providing specific and valuable research papers, articles, and academic resources. Search results are either irrelevant or overwhelming, making it challenging for researchers, students, and educators to locate the information they need. There is a critical need for a search engine that can modify search results to understand the semantic context of academic queries, and handle the ever-growing volume of academic data effectively. Furthermore, An Intelligent search function aims to understand the intent of user search and hence recommend more suitable relative output, to do so, this article is in charge for return keywords (Noun, Year, and Adj., etc.) from user input queries to removing user mistakes, Stop Words, etc. as well as understanding the user intention based machine learning techniques for the next work of this project.*

1. Related Work:

Numerous existing systems and academic research provide valuable reference points for our project. Well-known academic search engines like Google Scholar and Microsoft Academic serve as baselines for performance evaluation. Semantic Scholar, with its NLP-driven semantic search capabilities, offers insights into similar approaches. Research on recommender systems tailored for academic networks sheds light on personalized content recommendation. Natural language processing and machine learning are two approaches that are combined in intelligent search. For instance, it can create relations between semantic terms that an outdated search engine (one that's just considering at keywords) would be not capable to identify. It can similarly do "thinking" types of things like realize the structure of a document. The intersection of NLP and information retrieval provides guidance on applying natural language processing techniques in our search engine design. Recommended systems are generally three types and every method has their advantages and disadvantages such as Collaborative filtering, Content-based, Hybrid Method [1]. In this study, data attributes fall into two categories: structured attributes and unstructured attributes. The focus is on unstructured data, specifically the content of articles. Unstructured data typically requires conversion into structured data for use. In the context of an intelligent search engine module, the structures of academic papers are generally consistent, but their titles vary widely, and authors have unique writing styles. To really address and figure out the equivalence between these papers, the title of the paper should be provided in structured components. In paper recommender systems, a number of methodologies to demonstrating items have been proposed, one of which is the TF-IDF model [2]. TF-IDF (Term Frequency-Inverse Document Frequency) is a procedure developed in semantic hunting to determine the impact of words or expressions in a variety of reports, and it has been much of the time applied for data recovery and text mining [1]. It computes the recurrence of terms inside a report (TF) and their unique case across the whole record set (IDF). In semantic chasing, TF-IDF can be stretched out to think about terms and mixtures of words, assisting with gathering the importance of longer expressions. It upgrades indexed lists by arranging records in view of their consequence to the question, with those having significant terms or expressions receiving higher scores. Cosine similarities is a mathematical measure in the semantic search that evaluates how equivalent two reports depend on their element. It works out the cosine of the point between vectors addressing the records in a high-covered space. While larger angles point to dissimilarity, smaller angles specify more similarity. When applied to semantic detection, it's utilized to rank records in light of their likeness to a client's question. High-level semantic methods can be added to work on their viability in figuring out the significance and setting of words and expressions. Finally, the development of our system's user profiling and recommendation modules is guided by research on modified search engines and personalization in information retrieval.

2. **Implementation Challenges:** Making a NLP-powered document search engine has a bunch of intense parts. To jump with, the search engine necessities are essential to truly comprehend what the words mean in the archives. Then, entity acknowledgment represents a new obstacle, demanding the advancement of strong procedures to be familiar with and isolated substances, for example, author names and paper titles. Ensuring it understands why words matter in a sentence is as well challenging for more exact searches. The engine must resemble a language detective, figuring out how words are accompanying with one another. The selection of proper algorithms, optimization for processing large datasets, the setting up of metrics and approaches for evaluating performance, and guaranteeing a seamless user interface that maximizes the benefits of NLP without negotiating user experience all play a role in the technical integration of NLP into the architecture of the search engine.

3. **Problem Statements:** Present search engines face challenges in precisely understanding complex user queries, especially those including specific requests for structured data. For example, when a user inputs a query like "*I want Object Data model Structures */\ for Multimedia Data Typese by Frank Manola in 1990*," existing systems struggle to extract the crucial elements efficiently. This limitation hampers precise information retrieval, impacting the user's capability to achieve structured data. There is a pressing need to develop an intelligent search engine module with Natural Language Processing capabilities that can accurately decrypt such queries and return relevant information in a structured format, addressing the specific user mistakes like "Typese", or user writing slip (*/\) is also an important task.

4. **Methodology:** To address these issues, our methodology encompasses several key stages. Since we used a comprehensive preprocessed DBLP shortened dataset of academic papers including paper title, author name, and year. This data will be used to create a structured corpus, enriched with metadata such as author names, publication year, and keywords. Natural Language Processing (NLP) techniques will play a pivotal role in further data preprocessing, encompassing tasks like text normalization, tokenization, stemming, and named entity recognition to identify important entities such as authors, institutions, and research topics called keywords. Utilize TF-IDF to weigh the importance of terms in the corpus. This technique enhances the representation of documents and queries, giving more weight to terms that are rare in the corpus but significant in the context of the document or query. Later, calculate the cosine similarity between the TF-IDF representations of queries and documents. This measure will be used to rank and retrieve documents based on their relevance to user queries. By integrating TF-IDF and Cosine Similarity into the methodology, the search engine will enhance its document representation and retrieval capabilities, leading to more accurate and relevant search results for users in academic networks.

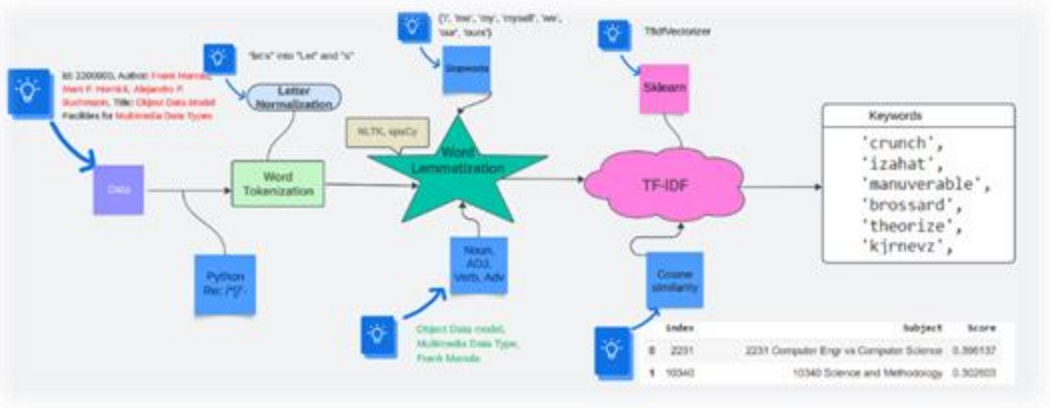

Fig. 1: workflow of finding keywords by NLP Techniques

5. **Dataset:** We are working with the DBLP dataset, which intentionally introduces a bit of noise to simulate real-life scenarios. The dataset has various versions and is notably large in scale. Our focus in this project is on a subset of the DBLP dataset, containing four fields: post ID, paper

title (ptitle), author information, and the year of publication. For the purpose of this part of project, our primary interest lies in the ptitle, author, and year metadata, which serve as our source of keywords. To streamline the data, we have combined the body, title, and year into a single field. Our dataset comprises 99,997 documents, each represented as a row, with a total of three columns. Additionally, our team has provided an actual shorter version of the DBLP dataset, consisting of 100,000 rows and 5 columns, where some data points have been excluded to enhance the dataset's clarity.

5. Experiments and Results: Machine Learning employs algorithms to train models for diverse tasks, ranging from text classification to question answering. Intelligent search capabilities, integral to a search engine, involve understanding user intent based on input. This is achieved through technologies like natural language processing (NLP) and machine learning, enhancing the engine's comprehension of user queries. To assess our intelligent search engine module, we conducted experiments. Investigating keyword extraction in academic networks is a well-explored area, with research on both sides of community structure studies [3] [4] [5]. In this project, we applied NLP and NLU skills, utilizing machine learning algorithms alike TF-IDF model, and cosine metrics for evolutionary determinations.

5.1 Data Process: In the domain of Natural Language Processing (NLP), a few principal strategies assume vital parts in refining the comprehension and processing of human language. Feeding the algorithm data still need to be treated for additional work. Tokenization, the underlying step, includes sorting out the text into individual words or tokens and working with the resulting analysis. Grammatical form labeling marks words with their linguistic jobs, recognizing things, action words, and modifiers. Names, places, and dates are examples of entities that are acknowledged and categorized by Named Entity Recognition (NER). Statistical models are used in language modeling to predict and generate human-like text, contributing to contextually proper responses. Stemming and Lemmatization further refine tokens, breaking them down for comparison, while plurals are normalized to their singular forms. Typos and spelling discrepancies are addressed through Typo Tolerance and Spell Check, ensuring robust search accuracy. Context Analysis involves understanding the context of a conversation to provide relevant responses. Semantic Parsing transforms natural language queries into structured representations, enhancing computational comprehension. Intent Recognition is integral, identifying the user's purpose behind a text or speech input, thereby contributing to a more nuanced and context-aware language processing system.

```
[ ] df_clean_ptitle['ptitle_author_year'][0]

'ontology,hydra,middleware,ambient,intelligent,device,peter,kostelnik,martin,sarnovsk
y,jan,hreno,2009'
```

Fig. 2: A paper title after applied all NLP techniques

5.2 Build Tf-Idf: Tf-Idf is a generally used NLP model that assists us determine the most significant words in each document in the corpus. We have used the CountVectorizer to create a vocabulary from all the papers in our dataset and generate counts for each row and the vocabulary size is "205563" meaning we have `205563` unique words (the columns) in our dataset minus the stopwords. We limit the size of the dictionary by setting `max_features=0.85 in some of the text mining applications, such as clustering and text classification when instantiating CountVectorizer. Then, we trained TF-IDF by using TfidfVectorizer from sklearn.

	Keyword	TF-IDF Score
162870	base	1255.389771
69973	system	1007.648132
164035	use	971.849182
1985	2015	901.400269
9900	2011	850.370789
...	...	...

Fig. 3: Keywords Extractions by TF-IDF

5.3 Semantic Search: Since, we have trained tfidf which returned us each keywords score. Once the index of keywords has been loaded and initialized, we can use it to locate papers that are based on related terms. When we enter a search query, the model will return the names of related publications along with the "Cosine Score," or similarity score. The more related the query to the document at the certain index demonstrating the higher similarity score. Below is the return results of my implementation.

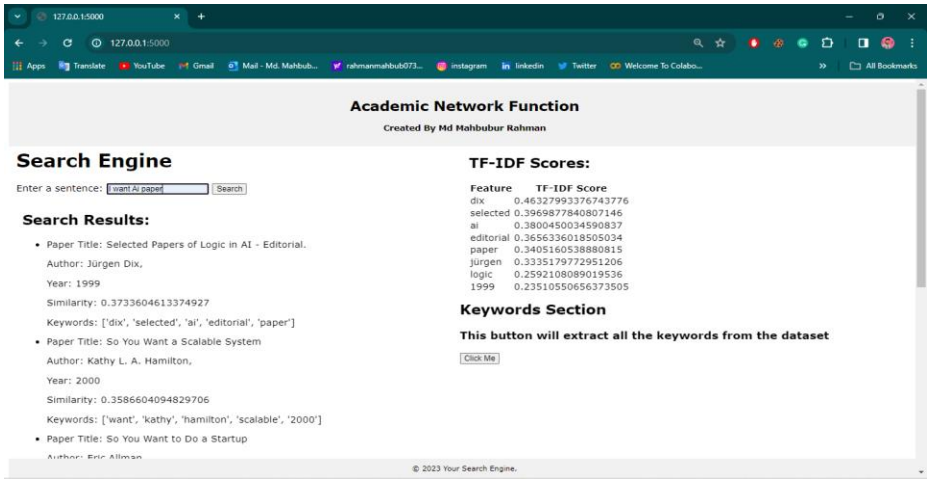

Fig. 4: Paper searching result by keywords in API

6. Implementation Tools: NLTK and SpaCy both provide user-friendly interfaces to more than 50 corpora and lexical resources such as WordNet, together with a collections of text processing libraries for classification, parsing, semantic reasoning, wrappers, part-of-speech tagging, tokenization, stemming, tagging, and Contextual spell checking, etc. In our project, we utilize the TfidfVectorizer class from the Scikit-Learn library to compute TF-IDF values for text data. The TfidfVectorizer efficiently transforms raw text into TF-IDF representations, enhancing our ability to analyze textual content by assigning weights to terms. Furthermore, we employ the cosine similarity function from Scikit-Learn to calculate the similarity between vectors, specifically those representing TF-IDF values. This approach allows us to quantify the similarity between different text elements. The project extensively utilizes Python programming language libraries, including but not limited to NumPy and Pandas. These libraries are instrumental in various data handling and processing tasks. Additionally, Jupyter Notebooks play a pivotal role by providing an interactive computing environment, seamlessly integrating code execution, visualization, and documentation.

7. Application Design: To analyzing this functionality, a web api has been made by using python flask api. Flask API simplifies communication between software applications, fostering modularity and scalability. With defined endpoints, it enables easy interaction, separates frontend from backend, and making it a go-to framework for building efficient and interoperable applications.

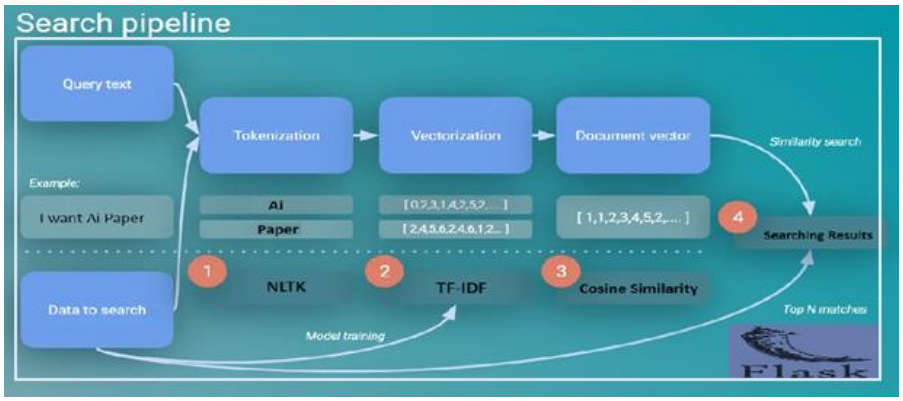

Fig. 5: API Architecture (Only for this Fragment)

8. **Conclusion:** In this work, an intelligent search module in academic with the possibility of recommendations for the user was developed. Different types of algorithms have their benefits and weaknesses. Thus, the plan was able to show the recommendations compared to existing facilities more accurately. Based on these studies, ML algorithm was chosen because it corresponds to words, additions, replacement of letters in the word. Technical means and technologies were chosen to solve the above goals and justified the feasibility of their use. An intelligent search system has been developed by using NLP algorithms. Extracted keywords have been executed for the subsequent work of this academic network project.

References

[1] P. R. a. H. S. C. D. Manning, ""Information Retrieval,"," *Stanford University,* 2008.

[2] D. G. a. N. X. M. Palmer, "Natural Language Processing: A Primer," *Morgan & Claypool Publishers,* 2010.

[3] Y. B. a. A. C. Goodfellow, Deep Learning, MIT Press, 2016.

[4] C. C. Aggarwal, Recommender Systems, Springer, 2016.

[5] D. A. a. J. Hendler, "Semantic Web for the Working Ontologist," *Elsevier,* 2011.

Source Code: *https://github.com/rahmanmahbub073/Search-Engine*