Identifying similarities of research papers utilizing Natural Language Processing and Information Retrieval methods

AYESHA SIDDIQUA and KHALID HASAN*, Missouri State University, USA

This project aims to create a system that helps researchers find similar research papers using advanced computer techniques. The system can show which papers are related by analyzing the words and ideas in the documents. This project focuses on developing a system that can automatically identify similarities between research papers by leveraging natural language processing (NLP) and information retrieval (IR) techniques. By analyzing the content of research papers, the system will provide insights into the relatedness of topics, methodologies, and findings across a large corpus of academic literature. The system will be easy to use, with a simple interface that anyone can understand. It will help researchers quickly find relevant papers for their work without spending plenty of time searching. Our project aims to make it easier for researchers to discover new information and learn from existing research. This project can potentially improve how research is done and help scientists make discoveries faster.

1 RELATED WORK

Recently, there has been significant research into effectively identifying similarities among documents [1, 6]. The survey by John Doe et al. [4] provides an overview of various text similarity techniques used in information retrieval, including traditional methods and recent advancements in NLP. Detecting similarities between research documents has a plethora of practical applications. For instance, two conferences might want to compare their submissions to identify similarities without revealing the real documents [5].

Along with representing research documents in a suitable data structure, measuring the similarity of research papers is a specific aspect of efficiently detecting document similarity. Currently, the Vector Space Model is the leading technique for document representation [9]. Each document is expressed as a weighted high-dimensional vector, the dimensions corresponding to individual features such as words [2]. Once the documents are structured in a suitable data format, similarity metrics like cosine similarity are applied to compute the documents' similarity scores. Wilson et al. [10] provide an overview of semantic similarity measures used in NLP tasks, including techniques based on ontology, distributional semantics, and deep learning. On top of that, a survey explores the use of word embeddings in information retrieval tasks, such as document similarity computation, and discusses their impact on retrieval performance [3]. Multiple similarity metrics and their effectiveness have been compared in Taghva et al. in [8].

Authors' address: Ayesha Siddiqua, as995s@missouristate.edu; Khalid Hasan, kh597s@missouristate.edu, Missouri State University, Springfield, Missouri, USA.

There has been ongoing research work on employing advanced research techniques to capture semantic relationships between research papers. Smith et al. [7] have reviewed the application of deep learning models for document similarity analysis, highlighting their effectiveness in capturing semantic relationships between documents. A recent study [11] introduces a natural language understanding-based method for automatically modeling project-specific property concepts from Binary Independence Models (BIM), demonstrating superior performance in ontology alignment and classification tasks.

2 PROPOSED METHOD

2.1 Data Collection

We have collected 24 research papers from *Google Scholar* [1] based on 6 different topics. The detail of our dataset is summarised in Table 1.

Topic	**Num. Papers**	**Paper Names**
Large Language Model(CS)	6	llm[1-6]
Mitochondria Research(BIO)	4	mitochondria[1-4]
Clustering Technique(CS)	4	clustering[1-4]
Cyber Security(CS)	3	cybersecurity[1-3]
Reinforcement Learning(CS)	4	reinforcement_learning[1-4]
Cancer Research(BIO)	3	cancer[1-3]

Table 1. Reseach Papers Across Diverse Research Topics

2.2 Data Preprocessing

After Collecting the research papers, we started preprocessing them. The papers are in pdf format. We extracted text from the pdf files using the *PyPDF*2 python library [2]. Next, we have removed the header and footer information from each page of the papers.

Moreover, we followed standard text processing steps for extracting interesting and non-trivial knowledge from unstructured text data. Our preprocess steps are as follows:

(1) Tokenize a document
(2) Convert tokens to lowercase
(3) Remove punctuations and stopwords
(4) Stem the tokens

[1] https://scholar.google.com/

[2] https://pypi.org/project/PyPDF2/

2.3 Information Retrieval Model

To identify similarities in research papers, we need to count the term frequency of each term in the vocabulary. Because of this crucial reason, we have considered the Vector Space Model (VSM) as a relevant data representation way for our project. The VSM is a widely used information retrieval model representing documents as vectors in a high-dimensional space, where each dimension corresponds to a term in the vocabulary. The VSM is based on the assumption that the meaning of a document can be inferred from the distribution of its terms and that documents with similar content will have similar term distributions.

We process our text data before constructing this model. Then, A term-document matrix is implemented, where each row represents a term and each column represents a document. The matrix contains the frequency of each term in each document, or some variant of it (e.g., term frequency-inverse document frequency, TF-IDF).

2.4 Similarity Computation

After constructing our model, we aim to reveal the closest documents as accurately as possible. To calculate this distance measurement, we experiment with two types of well-known measurements: Euclidian and Cosine similarity.

We apply cosine similarity for all three variants of term frequency: Term frequency (TF), Term/Inverse document frequency (TF-IDF), and Weight/Inverse document frequency (WF-IDF). By applying similarity measurements, we get the closest documents in pairs in sorted order.

We also want to uncover the clusters of documents, to be more specific, the documents are closest to each other in a cluster and the documents in different clusters are dissimilar. We apply both Euclidian and cosine similarity measures to find the relevant clusters in this case.

3 RESULTS

3.1 Comparison Of three IR techniques

In our Vector space model, we calculated three different Information retrieval techniques. Firstly, we calculated the term frequency of each unique term in each document. Next, we calculated the TF-IDF. The TF-IDF of each term in the documents is calculated using Equation 1.

$$\begin{gathered} TF - IDF_{t,d} = TF_{t,d} \times idf_t \\ idf_t = log_{10}(\frac{N}{df_t}) \end{gathered} \tag{1}$$

Here, N is the total number of documents in the collection. df_t is the document frequency of each term and $TF_{t,d}$ is term frequency of each term in each document. Finally, we have calculated sublinear TF scaling using Equation 2.

$$
\begin{gathered}
WF - IDF_{t,d} = WF_{t,d} \times idf_t \\
WF_{t,d} = \begin{cases} 1 + log_{10}(TF_{t,d}) & \text{if } TF_{t,d} > 0 \\ 0 & \text{Otherwise} \end{cases}
\end{gathered}
\tag{2}
$$

Finally, we have calculated the cosine similarity of each unique pair of documents. The cosine similarity is calculated using Equation 3.

$$
cos_{sim}(d1, d2) = \frac{d1 \times d2}{\|d1\| \times \|d2\|} \tag{3}
$$

We summarize the top 10 closest document pairs in Table 2. The table provides a comparative view of document similarities obtained using different weighting schemes. We witness fewer similarity scores using TF (i.e. maximum of 0.23) compared to other scoring measures TF-IDF and WF-IDF (i.e. maximum of 0.8 and 1.0, respectively). This illustrates the reasoning that TF-IDF and WF-IDF prioritize documents having less frequent and more distinctive terms. This implies that we have a more accurate assessment of document similarity in comparison with the basic TF scheme.

Using TF		**Using TF-IDF**		**Using WF-IDF**	
Doc Pair	**Similarity Score**	**Doc Pair**	**Similarity Score**	**Doc Pair**	**Similarity Score**
cancer1, cancer2	0.23	clustering2, clustering3	0.8	cancer1, cancer2	1.0
clustering2, clustering3	0.18	clustering3, custering4	0.72	clustering2, clustering3	1.0
llm3, llm5	0.14	cancer1, cancer2	0.71	clustering2, custering4	1.0
clustering2, custering4	0.13	clustering2, custering4	0.67	cancer1, cancer3	0.99
llm4, llm6	0.12	cancer1, cancer3	0.45	cancer1, cybersecurity2	0.99
clustering2, cybersecurity2	0.11	llm3, llm5	0.45	cancer1, llm1	0.99
clustering3, custering4	0.11	mitochondria1, mitochondria2	0.44	cancer1, llm5	0.99
clustering3, cybersecurity2	0.11	llm4, llm6	0.4	cancer1, mitochondria2	0.99
mitochondria2, mitochondria3	0.11	mitochondria2, mitochondria3	0.4	cancer2, llm1	0.99
mitochondria2, mitochondria4	0.11	mitochondria2, mitochondria4	0.38	cancer2, llm5	0.99

Table 2. Closest document pairs using TF, TF-IDF, and WF-IDF

3.2 Clustering

We have applied the k-means clustering algorithm to the documents. Since we have selected six different categories of research papers in our dataset, we chose cluster number six. First, we used the default distance measure as the Euclidean distance of the VSM using the TF-IDF scheme. The clustering results are presented in Table 3.

Label	Num. Papers	Documents
0	18	cancer3, clustering1, clustering3, cybersecurity2, cybersecurity3, cybersecurity4, llm1, llm3, llm4, llm6, mitochondria1, mitochondria2, mitochondria3, mitochondria4, reinforcement_learning1, reinforcement_learning2, reinforcement_learning3, reinforcement_learning4
1	1	custering4
2	1	llm5
3	2	cancer1, cancer2
4	1	llm2
5	1	clustering2

Table 3. K-Means Clustering results utilizing Euclidean Distance as distance measure

Here, we can observe that 18 research papers are clustered in the first cluster labeled as 0, which is not even close to the actual scenario. Other generated clusters consist of 1 or 2 papers. This observation infers that Euclidean distance is not a suggested parameter to cluster research papers. Next, we updated the distance measurement to cosine similarity and our experiment outcomes are displayed in Table 4.

Label	Num. Papers	Documents
0	6	llm1, llm2, llm3, llm4, llm5, llm6
1	4	mitochondria1, mitochondria2, mitochondria3, mitochondria4
2	3	clustering2, clustering3, custering4
3	3	cybersecurity2, cybersecurity3, cybersecurity4
4	5	clustering1, reinforcement_learning1, reinforcement_learning2, reinforcement_learning3, reinforcement_learning4
5	3	cancer1, cancer2, cancer3

Table 4. K-Means Clustering results utilizing Cosine similarity as distance measure

The K-means clustering using cosine similarity shows near-perfect results for the given dataset. We can see that, all 6 Large Language Model papers are clustered into Label-0, all 4 Mitochondria Related research papers are clustered together, and so on. The only wrong-labeled clustering is that it clustered one clustering paper with the reinforcement learning papers labeled with 4. As both are in the field of computer science related field, we assume that they have a large similarity measure. Our experiment results implicate that using cosine similarity to measure scores in the case of clustering research papers is one of the most effective ways.

4 CONCLUSION

In this project, we have calculated the similarity among research papers using different Information retrieval and machine learning techniques. From our experimental results, we can say that TF-IDF measurement with clustering algorithm using cosine similarity measurement shows great performance in identifying similar research papers. In the future, we can add a large number of research papers on diverse topics to our dataset which will create unique challenges to the given problem. Moreover, we can add various feature selection techniques like dimensionality reduction which will result in more accurate results.

REFERENCES

[1] Bassma S Alsulami, Maysoon F Abulkhair, and Fathy E Eassa. 2012. Near duplicate document detection survey. *International Journal of Computer Science and Communications Networks* 2, 2 (2012), 147–151.

[2] Fabio Benedetti, Domenico Beneventano, Sonia Bergamaschi, and Giovanni Simonini. 2019. Computing inter-document similarity with context semantic analysis. *Information Systems* 80 (2019), 136–147.

[3] Emily Brown et al. 2017. Enhancing Information Retrieval with Word Embeddings: A Survey. *Information Retrieval Review* 15, 1 (2017), 89–105.

[4] John Doe et al. 2018. A Survey of Text Similarity Approaches in Information Retrieval. *Journal of Information Retrieval* 12, 3 (2018), 245–269.

[5] Shahabeddin Geravand and Mahmood Ahmadi. 2014. An efficient and scalable plagiarism checking system using bloom filters. *Computers & Electrical Engineering* 40, 6 (2014), 1789–1800.

[6] Saul Schleimer, Daniel S Wilkerson, and Alex Aiken. 2003. Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. 76–85.

[7] Jane Smith et al. 2020. Deep Learning for Document Similarity Analysis: A Review. *Neural Networks* 30, 4 (2020), 567–589.

[8] Kazem Taghva and Rushikesh Veni. 2010. Effects of similarity metrics on document clustering. In *2010 Seventh International Conference on Information Technology: New Generations*. IEEE, 222–226.

[9] Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research* 37 (2010), 141–188.

[10] Michael Wilson et al. 2016. Semantic Similarity Measures for Natural Language Processing. *Natural Language Processing Journal* 8, 4 (2016), 321–345.

[11] Mengtian Yin, Llewellyn Tang, Chris Webster, Xiaoyue Yi, Huaquan Ying, and Ya Wen. 2024. A deep natural language processing-based method for ontology learning of project-specific properties from building information models. *Computer-Aided Civil and Infrastructure Engineering* 39, 1 (2024), 20–45.