DOCUMENT SIMILARITY USING TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY REPRESENTATION AND COSINE SIMILARITY METRIC

[1]Tarisa Januardini*, [2]Moch Ripal Malik Ababil

[1,2] Faculty of Science and Informatics, Universitas Pertiba

*Corresponding Author:
tarissa.januardini@gmail.com

Abstract

Document similarity is a fundamental task in natural language processing and information retrieval, with applications ranging from plagiarism detection to recommendation systems. In this study, we leverage the term frequency-inverse document frequency (TF-IDF) to represent documents in a high-dimensional vector space, capturing their unique content while mitigating the influence of common terms. Subsequently, we employ the cosine similarity metric to measure the similarity between pairs of documents, which assesses the angle between their respective TF-IDF vectors. To evaluate the effectiveness of our approach, we conducted experiments on the Document Similarity Triplets Dataset, a benchmark dataset specifically designed for assessing document similarity techniques. Our experimental results demonstrate a significant performance with an accuracy score of 89.53%. However, we observed instances where false predictions occurred due to paired documents having similar terms but differing semantics, revealing a weakness in the TF-IDF approach. To address this limitation, future research could focus on augmenting document representations with semantic features. Incorporating semantic information, such as word embeddings or contextual embeddings, could enhance the model's ability to capture nuanced semantic relationships between documents, thereby improving accuracy in scenarios where term overlap does not adequately signify similarity

***Keywords**: Document Similarity, TF-IDF, Cosine Similarity, Natural Language Processing, Information Retrieval*

1. INTRODUCTION

Document similarity serves as a fundamental cornerstone in numerous applications within natural language processing and information retrieval. Its significance spans from aiding in plagiarism detection to facilitating recommendation systems. Accurately quantifying the similarity between documents enables a deeper understanding of their content and relevance, thereby enhancing the efficiency and effectiveness of various computational tasks.

We focus on leveraging the term frequency-inverse document frequency (TF-IDF) approach, a widely used technique for document representation. TF-IDF allows us to transform textual data into high-dimensional vector representations, wherein the frequency of terms is weighted by their importance across documents while simultaneously mitigating the influence of common terms that may not be indicative of the document's content.

Subsequently, we employ the cosine similarity metric as a measure of similarity between pairs of documents. This metric evaluates the cosine of the angle between the TF-IDF vectors of two documents, providing a robust measure of similarity that is insensitive to variations in document length and term frequency distributions. By assessing the angle between these vectors, cosine similarity enables us to quantify the degree of similarity between documents, thereby facilitating tasks such as document clustering, retrieval, and categorization.

Our main objective is to evaluate the performance of a document similarity model that leverages TF-IDF representation and the cosine similarity metric. Through evaluation, we aim to determine the effectiveness of this approach in accurately quantifying document similarity and identify areas for improvement in current methodologies.

In this study, section 2 contains related works for TF-IDF, cosine similarity, and document similarity using document representation. Section 3 contains the method to the implemented system, and the components of the system. Section 4 contains the test that has been done and the evaluation from the results. Section 5 is the conclusion from the result and future improvements.

2. RELATED WORKS

TF-IDF is a widely used technique in information retrieval and text mining for representing documents as numerical vectors. It aims to capture the importance of terms in a document within a corpus (Rajaraman and Ullman, 2011). The TF-IDF value of a term in a document is calculated by multiplying its term frequency (TF) with the inverse document frequency (IDF). Term frequency represents the frequency of a term within a document, while inverse document frequency measures how unique a term is across the entire corpus. By weighing terms based on their frequency within a document and their rarity across the corpus, TF-IDF enables the identification of key terms that characterize the content of a document while downplaying common terms that may not be informative.

$$\mathrm{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

$$\mathrm{IDF}(t,D) = \log\left(\frac{\text{Total number of documents in the collection } |D|}{\text{Number of documents containing term } t}\right)$$

$$\text{TF-IDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D)$$

Cosine similarity is a metric used to measure the similarity between two vectors in a high-dimensional space. In the context of document similarity analysis, TF-IDF vectors are often utilized to represent documents, with each dimension corresponding to a unique term in the vocabulary. Cosine similarity computes the cosine of the angle between two TF-IDF vectors, providing a measure of similarity that ranges from -1 (perfectly dissimilar) to 1 (perfectly similar). This metric is particularly useful for comparing documents because it is invariant to the scale of the vectors and accounts for both the magnitude and direction of the vector representations. Documents with similar content will have TF-IDF vectors that point in similar directions, resulting in a higher cosine similarity score, whereas documents with dissimilar content will have TF-IDF vectors pointing in orthogonal directions, yielding a lower cosine similarity score.

In document similarity tasks, there are some document representations that are already used. These representations are word-based like bag of word, Latent Dirichlet Allocation

$$\text{cosine\_similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \times \|\mathbf{B}\|}$$

$$\mathbf{A} \cdot \mathbf{B} = \sum_{i=1}^{n} A_i \times B_i$$

$$\|\mathbf{A}\| = \sqrt{\sum_{i=1}^{n} A_i^2}$$

(LDA), and paragraph vectors (Dai et al., 2015). Dai et al. used paragraph vectors as representation to document similarity based on clustering of Wikipedia English articles. Then, the document distance is calculated by cosine similarity. The proposed approach tested on hand-built triplets of Wikipedia articles dataset. The dataset consists of 172 triplet's URL of English Wikipedia article, or 516 documents in total. The test gives the result prediction accuracy 93% for paragraph vector, LDA 82%, and bag of words 86%.

3. PROPOSED APPROACH

The dataset used to test our proposed method is hand-built triplets of Wikipedia articles made by Dai et al. The dataset consists of 172 triplets document URL from English Wikipedia articles that are labeled by 1 of the 27 categories (ex: machine learning, books, places, animals, etc.). There are some categories that need a deeper understanding of the document like places.

Triplet URL will predict as true if the distance between first and second document URL is closer (in cosine similarity context, is higher) than the distance between second and third document URL.

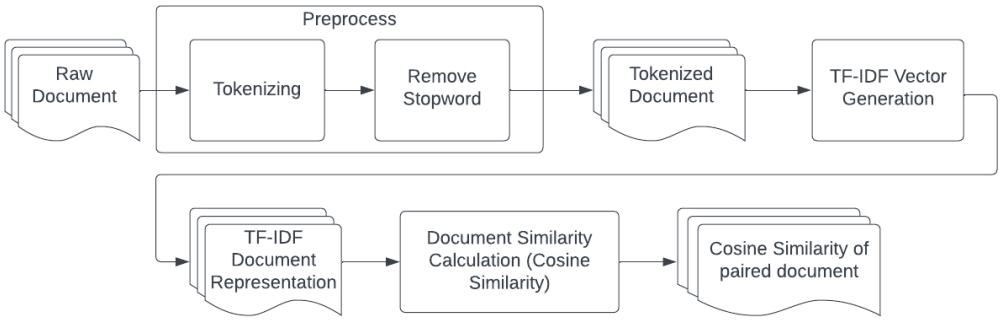

Figure 1. The Architecture of Implemented System

Preprocess is conducted to clean and reduce the noise in the raw document. Tokenizing and removing stop words process use python package NLTK. TF-IDF vector is generated using python package scikit-learn. The result is the vector representation of each document in corpus.

Cosine similarity calculation is implemented with python package scikit-learn. This process only calculated two pairs (first to second and second to third document) from each triplet.

4. RESULTS AND DISCUSSION

The proposed system gives the accuracy of 89.53%. The result is comparable with other word-based representations models.

Table 1. Comparison of proposed model to other method

Model	Accuracy
Paragraph Vector (Dai et al., 2015)	93%
LDA (Dai et al., 2015)	82%
Averaged word embedding (Dai et al., 2015)	84.9%
Bag of words (Dai et al., 2015)	86.0%
TF-IDF (Proposed Method)	**89.5%**

False predictions may arise when a document shares similar topics and terms, requiring a semantic approach to discern which document pair is more akin. The determination of which document pair exhibits greater similarity depends on the semantic context utilized for differentiation. An example of this is a triplet comparison involving articles “Java”, “C++”, and “C”. In the context of programming paradigms, "Java" and "C++" exhibit greater similarity, a categorization labeled as a true prediction by the dataset author. However, in terms of historical development and syntax, "C++" and "C" demonstrate more resemblance.

Table 2. Example of article text (snippet) from false predictions

Title	Text
Java	Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. ...
C++	C++ (, pronounced "C plus plus" and sometimes abbreviated as CPP) is a high-level, general-purpose programming language created by Danish computer scientist Bjarne Stroustrup. First released in 1985 as an extension of the C programming language, it has since expanded significantly over time; as of 1997 C++ has object-oriented, generic, and functional features, in addition to facilities for low-level memory manipulation. ...
C	C (pronounced – like the letter c) is a general-purpose computer programming language. It was created in the 1970s by Dennis Ritchie, and remains very widely used and influential. By design, C's features cleanly reflect the capabilities of the targeted CPUs. It has found lasting use in operating systems, device drivers, and protocol stacks, but its use in application software has been decreasing. C is commonly used on computer architectures that range from the largest supercomputers to the smallest microcontrollers and embedded systems. ...

5. CONCLUSION

In this study, a document similarity model for English documents is developed based on TF-IDF representation and cosine similarity metric. The model was tested using hand-built triplets of Wikipedia articles from Dai et al. The proposed method gives significant performance with an accuracy score of 89.53%. The result is comparable with other word-based representation models, but it still has to be improved. Therefore, suggestions for future research include how to improve the model such as adding a semantic-based approach as the representation to get the semantic context from the document.

NOVELTY

Document similarity model using TF-IDF as document representation and cosine similarity as metric for real world long document dataset.

References

A. Rajaraman and J. D. Ullman, "Data Mining," in Mining of Massive Datasets, pp. 1–17, 2011. doi: 10.1017/CBO9781139058452.002.

A. Singhal, "Modern Information Retrieval: A Brief Overview," Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, vol. 24, no. 4, pp. 35–43, 2001.

A. M. Dai, C. Olah, and Q. V. Le, "Document embedding with paragraph vector," arXiv preprint arXiv:1507.07998, 2015.