

Document retrieval using term frequency inverse sentence frequency weighting scheme

Mohannad T. Mohammed¹, Omar Fitian Rashid²

¹College of Health and Medical Technology, Middle Technical University, Baghdad, Iraq

²College of Science, University of Baghdad, Baghdad, Iraq

Article Info

Article history:

Received Oct 1, 2022

Revised Apr 29, 2023

Accepted May 6, 2023

Keywords:

Document representation

Document retrieval

Similarity measures

Term frequency inverse

sentence frequency

Weighting schemes

ABSTRACT

The need for an efficient method to find the furthestmost appropriate document corresponding to a particular search query has become crucial due to the exponential development in the number of papers that are now readily available to us on the web. The vector space model (VSM) a perfect model used in “information retrieval”, represents these words as a vector in space and gives them weights via a popular weighting method known as term frequency inverse document frequency (TF-IDF). In this research, work has been proposed to retrieve the most relevant document focused on representing documents and queries as vectors comprising average term term frequency inverse sentence frequency (TF-ISF) weights instead of representing them as vectors of term TF-IDF weight and two basic and effective similarity measures: Cosine and Jaccard were used. Using the MS MARCO dataset, this article analyzes and assesses the retrieval effectiveness of the TF-ISF weighting scheme. The result shows that the TF-ISF model with the Cosine similarity measure retrieves more relevant documents. The model was evaluated against the conventional TF-ISF technique and shows that it performs significantly better on MS MARCO data (Microsoft-curated data of Bing queries).

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Omar Fitian Rashid

College of Science, University of Baghdad

Baghdad, Iraq

Email: omaralrawi08@yahoo.com

1. INTRODUCTION

Searching through a repository of documents to find those that match a particular topic is known as information retrieval, and the effectiveness of the process depends on how accurately the data is retrieved. Document retrieval is a form of, text retrieval. Text retrieval is a subset of information retrieval in which information is basically kept in the form of text. Text retrieval is an important field of research nowadays because it is the foundation of all internet search engines [1]. Due to its well-defined statistical foundations and strong empirical performance, the language modelling technique for information retrieval has recently received much attention [2]. The best example of this application is web searches. Several algorithms have been designed for this purpose. They take an input query and compare it to the stored documents or text samples, ranking the results according to how similar they are to the given query [3]. These algorithms compare the individual query phrases to the indexed documents, which preserve data on term frequencies and locations. Each document receives a grade depending on its similarity to other documents. A high frequency of presence in a document results in a high query word score for that content [4]. Finding and ranking unstructured documents that correspond to the user's information demands is the aim of information retrieval [5]. When a user submits a search query to the system, an information retrieval process starts.

The information retrieval system uses a variety of models to comprehend the user's query. Following these models, it ranks all documents and displays the most pertinent ones from the data set [6]. Results from these several methods will be shown. Various algorithms employ various strategies to evaluate this similarity and determine the score. The vector space model (VSM) is among the most respected and widely applied methods. It defeats the Boolean model, which uses Boolean logic to match documents with queries, regardless of whether the necessary phrases are present in the document [7]. A popular model is the vector space model, which uses term frequency inverse document frequency (TF-IDF) as one of its weighting methods [8]. The number n represents how many terms are used to create an index to exemplify the documents. It describes text elements represented as vectors in n dimensions [9]. The document must be stripped and separated into different terms to create an index. Then, the document can be processed further, which condenses several word forms into a single stem, improving the efficiency of matching two papers. The term frequency inverse sentence frequency (TF-ISF) weighting technique [10], [11], which operates at the sentence level rather than the document level, is used in this research to enhance the retrieval of documents. Comparing the method with a baseline model (TF-IDF) will also show that a suggested retrieval system performs significantly better. It also looked at how different similarity metrics affected the model. The TF-IDF and TF-ISF schemes used Cosine and Jaccard as similarity measures. A basic method for information retrieval has been illustrated in Figure 1.

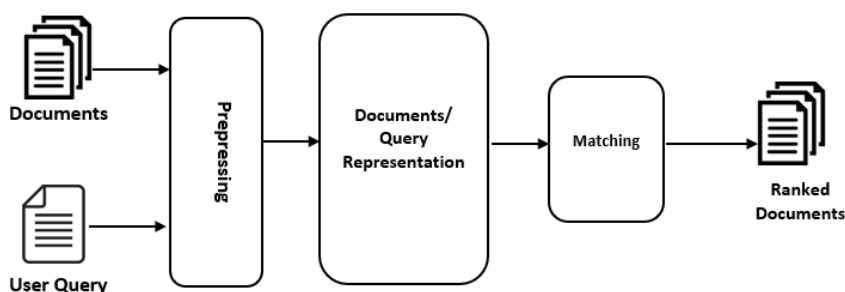


Figure 1. Document retrieval system

The user enters their query into the system while a collection of documents is given to the representation model in this image, which then executes the representing process. A matching algorithm then uses the user query to obtain the top- k -ranked documents [12]. To do this work, a predetermined representation model is provided, and training data is then subjected to matching algorithms and a rating model to match the user query. Then the result's page is obtained [13]. Before assigning ranks to the documents, the information retrieval system performs a few preprocessing steps that will be covered in the following section. The vector space model's TF/IDF weighting method has been replaced with a modified system for term weighting called term frequency with average term occurrences (TF-ATO) [14], which, as findings have demonstrated, is an enhancement. The method determines the average number of times a term appears in documents and uses discrimination to weed out the less important weights. When combined with stop word removal and the researchers' suggested discriminative strategy, TF-ATO outperforms the TF/IDF system. According to the document length, a different modified TF-IDF approach that includes "relative TF-weighting" and "TF-normalization" has been proposed [15]. Most available models use a poorly balanced single-term frequency normalization between favouring short and long documents for searches of various lengths. This weighting system makes use of two euclidean distances that cannot be used to calculate this proximity and produce accurate results.

The area where information is spread apart, approximation "Kolmogorov complexity" exists, Shirakawa *et al.* [16] have demonstrated that the distance between the term and the empty string is equal to the inverse document frequency (IDF) of a phrase. They have developed a potential development of IDF called N-gram IDF, which manages any length of words and terms depending on this discovery. Without utilizing natural language processing (NLP) approaches, this scheme finds that the dominating N-grams are used to separate overlapping ones and extract any length of key terms from texts. The weight of every feasible N-gram is determined using two string processing methods—this method used cutting-edge techniques for online search query partitioning and key term extraction to attain competitive performance. Three novel weighting techniques have been proposed in [17] to enhance the TF-IDF weighting technique. The first strategy, known as dispersed words weight augmentation (DWWA), gives words that are dispersed over the majority of the paragraphs in the document more weight than the words that are only found within some sections. The second method, known as title weight augmentation (TWA), gives the words in the document's title and opening

sentences greater weight. The third technique referred to as first ranked terms weight augmentation (FRWWA), gives terms that exist the most repeatedly in a document more weight.

2. RESEARCH METHOD

The proposed model's general structure is depicted in Figure 2. In this model, all documents in the corpus are segmented into individual sentences. After that, the sentences are preprocessed to make them simple for search engines and space and time requirements. The preprocessing steps in this stage include tokenization, punctuation elimination, removing duplicate tokens, stop-words removal, and stemming. Using the natural language toolkit (NLTK) library for Python, queries and documents were both preprocessed [18].

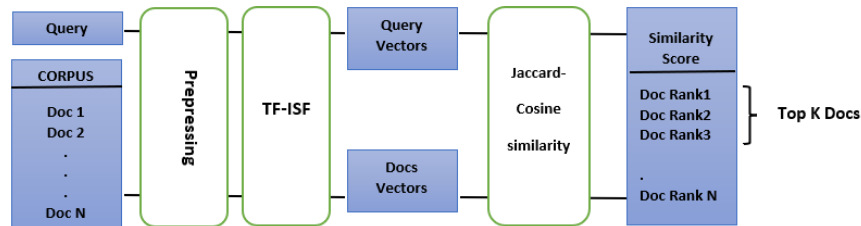


Figure 2. The general framework of the proposed model

2.1. Pre-processing

Pre-processing is adding a new document to an information retrieval system [19]. Each document must undergo various pre-processing procedures for search engines to effortlessly understand it as a document and process it using their algorithms. Each document is subjected to processes like ("text segmentation, tokenization, stemming, and stop word removal"). Figure 3 depicts the preprocessing phases, and a description of each step is provided in:

- Text segmentation: documents are divided into separate sentences using the delimiters ".", "?", and "!". The textual units required for comparing the query and corpus are these distinct sentences that come from the segmentation procedure.
- Tokenization: is the process of breaking up text into tokens using blank spaces as delimiters. Possibly while also removing specific characters, such as punctuation.
- Stemming: which replaces all word variants with the word's single stem or root, is described in [20]. In the majority of information retrieval (IR) systems, it is crucial. Stemming often involves stripping words of any added suffixes and prefixes. For instance, the words "reading," "reader," and "reads" are normalized to "read".
- Stop-words: although a document may have hundreds or even thousands of words, not every word is equally significant from the user's point of view. Typically, this word is used frequently throughout the text without adding anything to make it more informative. Practically speaking, stop words like "the," "a," and "of," and users do not search for numerous others. Eliminating these terms helps to improve the findings' accuracy and also enables a reduction in the amount of memory space needed. The English stop word list contains the extracted stop words [21]. Stop-word lists can varied for different document collections based on the objective.

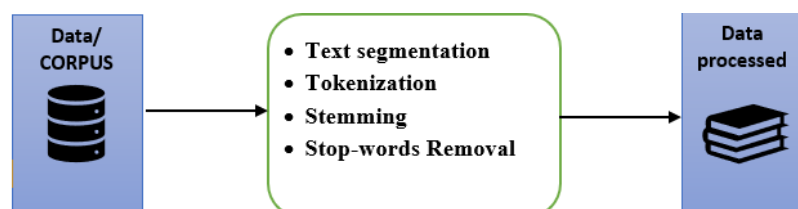


Figure 3. Preprocessing diagram

2.2. Document representation

There are various ways to model a text document for information retrieval. Fingerprinting is a standard heuristic retrieval model. Using fingerprinting, approaches represent a document by fragmenting it into substrings and choosing a subset of all the generated substrings. A document fingerprint leans towards recognizing the document uniquely as a human fingerprint does [22]. Another commonly used retrieval model is the VSM. VSM is a standard technique for traditional IR. It is also used in information filtering, relevancy rankings and indexing. It is a simple model representing each document as a vector of terms. If words are chosen as terms, then every word in the vocabulary becomes an independent dimension in a very high dimensional vector space. Any text can then be represented as a vector in this high-dimensional space [23]. The angle between two vectors is used as a measure of divergence between the vectors.

Weighting scheme: to select the most pertinent terms to verify, a term weighting scheme is an essential component of all vector space document models. Different weighted systems can represent a document as a collection of terms, such as the term frequency-inverse document frequency. Variations from one method to another belong to the specific choices of weights in the vectors.

TF-IDF weighting scheme: because they are straightforward to understand, VSMs are commonly implemented for text representation. Each term in VSM is given a weight based on a separate weighting scheme. The most popular weighing method, especially in IR, is TF-IDF [24]. TF-IDF intuitively assesses the significance of a specific term in a particular document. The TF component measures the frequency of a term in a particular document, normalized by the document's size, to eliminate any tendency toward more significant documents. IDF calculates how many documents in the corpus include the specified phrase. It represents the frequency or rarity of the phrase across the entire corpus. Combining these two statistics, a term is given more value if it frequently occurs in a text and less importance if it frequently occurs throughout the corpus [25]. Doing this gives keywords more weight and standard terms like articles and prepositions less weight. The formula for TF-IDF is as follows if t , and d , respectively, stand for terms and documents as in (1), (2) and (3) respectively:

$$TF(t, d) = \frac{\text{Count OF } t \text{ in } d}{\text{Number OF terms in } d} \quad (1)$$

$$IDF(t, d) = \log_e \frac{\text{Number Of documents}}{\text{Number of documents with } (t)} \quad (2)$$

TF-IDF is the product of the above two statistics.

$$TFIDF(t, d) = TF(t, d).IDF(t, d) \quad (3)$$

In this work representing documents and queries as vectors comprising average term TF-ISF weights instead of representing them as vectors of term RF-ISF as in (4), (5) and (6) illustrated in:

$$TF(t, s) = \frac{\text{Number of times } t \text{ appears in a sentence}}{\text{Number OF terms in } s} \quad (4)$$

$$ISF(t, s) = \log_e \frac{\text{Number Of sentences}}{\text{Number of sentences with } (t)} \quad (5)$$

the result of the two statistics above is TF-ISF.

$$TFISF(t, s) = TF(t, s).ISF(t, s) \quad (6)$$

2.3. Similarity measures

An essential part of text-related research and applications in many tasks, such as text categorization, text summarization, IR, document clustering, and others, is measuring the similarity of words, sentences, phrases, and documents [26]. Calculating similarity between words is an essential part of measuring similarity between texts which is used later as the main step for calculating similarities between sentences and documents. The similarity between words can be satisfied lexically and semantically. The lexical similarity between words can occur if they have a similar character sequence. Semantic similarity can occur if the terms have the same thing, used in the same context.

Two specific and effective similarity metrics are Cosine and Jaccard, which are utilized with the TF/ISF system. Jaccard similarity is computed as the number of shared terms over the number of all unique terms in both strings. The *Jaccard* similarity ranges between 0 and 1, where one means that the two objects are the same and 0 means they are completely different. As shown in (7), Jaccard similarity is calculated.

$$\text{Jaccard}(S1, S2) = \frac{S1 \cap S2}{S1 \cup S2} \quad (7)$$

Also, the Cosine similarity which measures the similarity between two vectors of an inner product space that calculates the Cosine of the angle between them where the same way represents the query as the documents described. Each dimension represents a term with its weight in the paper, which is non-negative. As a result, the Cosine similarity is non-negative and bounded between [0,1] as shown in Figure 4, then score can be computed between two sentences S1 and S2 (two TF-ISF vectors) as in (8).

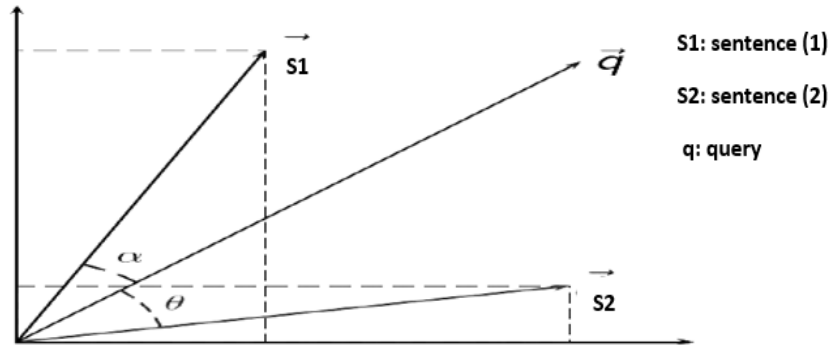


Figure 4. Cosine similarity with VSM

$$\text{Cosine}(S1, S2) = \frac{\sum_{k=1}^m \text{Tfifsf}_1 \cdot \text{Tfifsf}_2}{\sqrt{\sum_{k=1}^m \text{Tfifsf}_1^2 \cdot \sum_{k=1}^m w\text{Tfifsf}_2^2}} \quad (8)$$

Next, a relevance score is determined throughout the whole corpus, for each query-document pair. Finally, depending on this rating, the Top K documents are obtained.

3. RESULTS AND ANALYSIS

3.1. Dataset

A portion of the MS MARCO data was chosen to assess the efficacy of our strategy. With more than “one million queries and three million documents”, MS MARCO is a sizable data set designed for document re-ranking [27]. All of the queries were taken as samples from private Bing searches. For this study, a subgroup of 500 queries and the top 100 relevant documents for each query, for a total of 20,000 documents, were selected.

3.2. Metrics

Mean average precision (MAP) was used since each document's relevancy ranking for each query was marked in the “MS MARCO data” the following is a more thorough explanation of the MAP metric: MAP: precision is a measure of a model's complete prediction accuracy. The mean average precision value is determined for the model by averaging the precision values from all the queries as in (9).

$$\text{MAP}(k) = \frac{\text{relevant docs} \cap \text{retrieved docs}}{\text{retrieved docs}} \quad (9)$$

3.3. Results

A document was deemed relevant by the model for purposes of this metric if it was among the “top 10, 5, and 3 documents in the MS MARCO data” for each query. Since a user might only be interested in a limited subset of the obtained pages depending on the scenario, the model evaluated MAP scores on smaller slices in addition to retrieving ten documents per query by default. The studies performed using the recently developed methodologies are described in this section TF-ISF with Cosine similarity (TF-ISF-Co), and TF-ISF with Jaccard similarity (TF-ISF-Ja). It then compares their results with the Baseline TF-IDF weighting measure, Cosine (TF-IDF-Co) and Jaccard similarity (TF-IDF-Ja). The experiments have been conducted using a subset of the MS MARCO data as an evaluation dataset for evaluating the proposed system. Table 1 and Figure 5 show that TF-ISF trained on MS MARCO data outperformed Baseline TF-IDF. Also, obviously, the Cosine similarity measure is better than the Jaccard similarity measure in the case of the TF-IDF and TF-ISF weighting scheme.

Table 1. Evaluation results

IR-model	MAP(k=3)	MAP(k=5)	MAP(k=10)
TF-IDF-Ja (Baseline)	0.752	0.743	0.691
TF-IDF-Co (Baseline)	0.781	0.774	0.732
TF-ISF-Ja	0.761	0.756	0.713
TF-ISF-Co	0.818	0.804	0.755

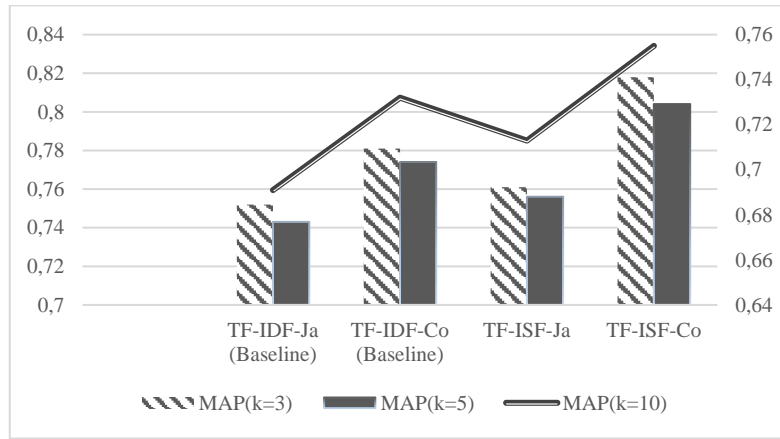


Figure 5. Evaluation results of the different models

The results show that the precision of the TF-ISF-Ja Model was better than the TF-IDF-Ja (Baseline). Also, it is noticeable when TF-ISF is using the Cosine similarity measure. According to Table 1, the precision value rises from 78% to 81%. Also, the relative improvement percentage (RIP) of the proposed IR models TF-ISF-Co and TF-ISF-Ja against the existing TF-IDF IR models in terms of MAP(k=3), MAP(k=5) and MAP(k=10) which have been evaluated using MS MARCO data as an evaluation dataset, is illustrated in the Table 2 and Figure 6. The relative improvement percentage is calculated according to (10).

$$RIP = \frac{\text{our method} - \text{other method}}{\text{other method}} \times 100 \tag{10}$$

Table 2. Relative improvement percentage of the proposed IR model against the Baseline

IR-model	Relative improvement %		
	MAP(k=3)	MAP(k=5)	MAP(k=10)
TF-ISF-Ja vs TF-IDF-Ja (Baseline)	1.20%	1.75%	3.18%
TF-ISF-Co vs TF-IDF-Co (Baseline)	4.74%	3.88%	3.14%

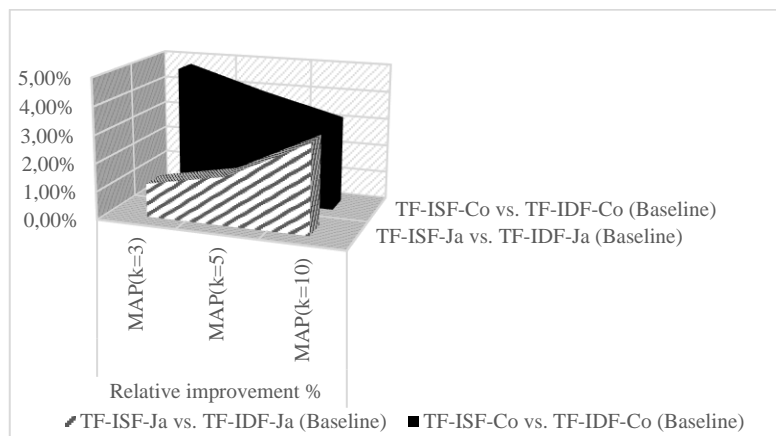


Figure 6. Relative improvement percentage of the proposed IR model against the Baseline

4. CONCLUSION

This work presented a new representation method for IR representing documents and queries as vectors of average term weight with the Cosine and Jaccard similarity measures, where the relatedness of a query to a given document is calculated based on the lexical relatedness of their concepts. The results of the studies demonstrated that the novel technique enhanced the information retrieval system's performance as measured by the MAP metric. Future work could go into several different areas. The first approach is the creation of new models, weighting techniques, and similarity measurements that can function well with the same semantic information. Second, the model's accuracy is increased by the creation of methods for combining lexical and semantic data. The usefulness of the hybrid retrieval strategy for other information retrieval tasks, such as question answering and recommendation systems, would also be interesting to assess.




REFERENCES

- [1] R. W. P. Luk, "Why is information retrieval a scientific discipline?," *Foundations of Science*, vol. 27, no. 2, pp. 427–453, Jun. 2022, doi: 10.1007/s10699-020-09685-x.
- [2] Z. A. Yilmaz, W. Yang, H. Zhang, and J. Lin, "Cross-domain modeling of sentence-level evidence for document retrieval," 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)*, 2019, pp. 3488–3494, doi: 10.18653/v1/D19-1352.
- [3] S. Hofstätter, H. Zamani, B. Mitra, N. Craswell, and A. Hanbury, "Local self-attention over long text for efficient document retrieval," in *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2020, pp. 2021–2024, doi: 10.1145/3397271.3401224.
- [4] D. K. Sharma, R. Pamula, and D. S. Chauhan, "A hybrid evolutionary algorithm based automatic query expansion for enhancing document retrieval system," *Journal of Ambient Intelligence and Humanized Computing*, Feb. 2019, doi: 10.1007/s12652-019-01247-9.
- [5] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "BEIR: a heterogenous benchmark for zero-shot evaluation of information retrieval models," *arXiv:2104.08663*, Apr. 2021.
- [6] J. Guo *et al.*, "A deep look into neural ranking models for information retrieval," *Information Processing and Management*, vol. 57, no. 6, 2020, doi: 10.1016/j.ipm.2019.102067.
- [7] S. S. Samant, N. L. B. Murthy, and A. Malapati, "Improving term weighting schemes for short text classification in vector space model," *IEEE Access*, vol. 7, pp. 166578–166592, 2019, doi: 10.1109/ACCESS.2019.2953918.
- [8] A. Jain, A. Jain, N. Chauhan, V. Singh, and N. Thakur, "Information retrieval using cosine and jaccard similarity measures in vector space model," *International Journal of Computer Applications*, vol. 164, no. 6, pp. 28–30, Apr. 2017, doi: 10.5120/ijca2017913699.
- [9] H. Zamani, S. Dumais, N. Craswell, P. Bennett, and G. Lueck, "Generating clarifying questions for information retrieval," in *The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020*, Apr. 2020, pp. 418–428, doi: 10.1145/3366423.3380126.
- [10] C. V. Gysel, M. D. Rijke, and E. Kanoulas, "Neural vector spaces for unsupervised information retrieval," *ACM Transactions on Information Systems*, vol. 36, no. 4, pp. 1–25, Oct. 2018, doi: 10.1145/3196826.
- [11] M. T. Mohammed, N. J. Kadhim, and A. A. Ibrahim, "Improved VSM based candidate retrieval model for detecting external textual plagiarism," *Iraqi Journal of Science*, vol. 60, no. 10, pp. 2257–2268, Oct. 2019, doi: 10.24996/ijcs.2019.60.10.20.
- [12] S. Marcos-Pablos and F. J. García-Peñalvo, "Information retrieval methodology for aiding scientific database search," *Soft Computing*, vol. 24, no. 8, pp. 5551–5560, 2020, doi: 10.1007/s00500-018-3568-0.
- [13] T. Russell-Rose, J. Chamberlain, and L. Azzopardi, "Information retrieval in the workplace: A comparison of professional search practices," *Information Processing and Management*, vol. 54, no. 6, pp. 1042–1057, 2018, doi: 10.1016/j.ipm.2018.07.003.
- [14] O. A. S. Ibrahim and D. Landa-Silva, "Term frequency with average term occurrences for textual information retrieval," *Soft Computing*, vol. 20, no. 8, pp. 3045–3061, Aug. 2016, doi: 10.1007/s00500-015-1935-7.
- [15] J. H. Paik, "A novel TF-IDF weighting scheme for effective ranking," in *SIGIR 2013 - Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2013, pp. 343–352, doi: 10.1145/2484028.2484070.
- [16] M. Shirakawa, T. Hara, and S. Nishio, "N-gram IDF: A global term weighting scheme based on information distance," in *WWW 2015 - Proceedings of the 24th International Conference on World Wide Web*, May 2015, pp. 960–970, doi: 10.1145/2736277.2741628.
- [17] J. Wang and Y. Dong, "Measurement of text similarity: A survey," *Information (Switzerland)*, vol. 11, no. 9, pp. 1–17, Aug. 2020, doi: 10.3390/info11090421.
- [18] M. Wang and F. Hu, "The application of nltk library for python natural language processing in corpus research," *Theory and Practice in Language Studies*, vol. 11, no. 9, pp. 1041–1049, Sep. 2021, doi: 10.17507/tpls.1109.09.
- [19] F. A. Ruambo and M. R. Nicholas, "Towards enhancing information retrieval systems: A brief survey of strategies and challenges," *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, vol. 2019-October, 2019, doi: 10.1109/ICUMT48472.2019.8970954.
- [20] I. Boban, A. Doko, and S. Gotovac, "Sentence retrieval using stemming and lemmatization with different length of the queries," *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 3, pp. 349–354, 2020, doi: 10.25046/aj050345.
- [21] M. Jaiswal, S. Das, and Khushboo, "Detecting spam e-mails using stop word TF-IDF and stemming algorithm with Naïve Bayes classifier on the multicore GPU," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 4, pp. 3168–3175, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3168-3175.
- [22] N. B. Unnam and P. K. Reddy, "A document representation framework with interpretable features using pre-trained word embeddings," *International Journal of Data Science and Analytics*, vol. 10, no. 1, pp. 49–64, Jun. 2020, doi: 10.1007/s41060-019-00200-5.
- [23] O. Shahmirzadi, A. Lugowski, and K. A. Younge, "Text similarity in vector space models: A comparative study," *SSRN Electronic Journal*, 2018, doi: 10.2139/ssrn.3259971.
- [24] F. Günther, L. Rinaldi, and M. Marelli, "Vector-space models of semantic representation from a cognitive perspective: A discussion of common misconceptions," *Perspectives on Psychological Science*, vol. 14, no. 6, pp. 1006–1033, Nov. 2019, doi: 10.1177/1745691619861372.




- [25] D. Gillick *et al.*, "Learning dense representations for entity retrieval," in *CoNLL 2019 - 23rd Conference on Computational Natural Language Learning, Proceedings of the Conference*, 2019, pp. 528–537, doi: 10.18653/v1/k19-1049.
- [26] S. Zhou, X. Xu, Y. Liu, R. Chang, and Y. Xiao, "Text similarity measurement of semantic cognition based on word vector distance decentralization with clustering analysis," *IEEE Access*, vol. 7, pp. 107247–107258, 2019, doi: 10.1109/ACCESS.2019.2932334.
- [27] P. Bajaj *et al.*, "MS MARCO: A human generated machine reading comprehension dataset," *arXiv:1611.09268*, Nov. 2016.

BIOGRAPHIES OF AUTHORS



Mohannad T. Mohammed    was born in Baghdad, Iraq in 1986. He got his B.Sc. in computer science and M.Sc. in computer science/NLP from the College of Science, University of Baghdad in 2010 and 2019. His main field of interest is the AI and natural language processing. He can be contacted at email: mohannad.tm@mtu.edu.iq.



Omar Fitian Rashid    was born in Baghdad, Iraq in 1988. He got his B.Sc. in computer science and M.Sc. in computer security from the College of Science, University of Baghdad in 2010 and 2014. He has successfully defended his Ph.D. thesis in 2019 entitled "Enhanced DNA encoding for anomaly intrusion detection system" at the Faculty of Information Science and Technology, National University of Malaysia (Universiti Kebangsaan Malaysia). His main field of interest is the computer and network security. He can be contacted at email: omaralrawi08@yahoo.com.