Jurnal Kejuruteraan 36(4) 2024: 1535–1543
https://doi.org/10.17576/jkukm-2024-36(4)-19

Efficient Document Retrieval System using Locality Sensitive Hashing Nearest Neighbor Algorithm and Weighted Jaccard Distance for Retrieving Closest Personalities

E. Ben George[a*], G. Jeba Rosline[a], N. Balasupramanian[b] & N.R.Wilfred Blessing[c]

[a]*IT Department, College of Computing and Information Sciences, University of Technology and Applied Sciences-Muscat, Sultanate of Oman.*

[b]*IT Department, Rajiv Gandhi Engineering College, Pondichery, India.*

[c]*IT Department, College of Computing and Information Sciences, University of Technology and Applied Sciences-Ibri, Sultanate of Oman.*

[*]*Corresponding Author: e.bengeorge@gmail.com*

Received 6 January 2024, Received in revised form 6 January 2024
Accepted 6 February 2024, Available online 30 July 2024

ABSTRACT

The process of retrieving significant documents based on the search key from a corpus has been a vital research problem in the information retrieval field. This paper proposes an efficient way to retrieve documents related to different personalities extracted from Wikipedia. The proposed method utilizes the Locality Sensitive Hashing Nearest Neighbor algorithm combined with Weighted Jaccard Distance to measure document similarity with enhanced precision. This document retrieval system demonstrates competitive performance compared to existing methods in the Personality Identification domain. The introduction of a document centroid normalization technique significantly improves the effectiveness of information retrieval by enabling better discrimination between documents. The personality document search results were compared for different distance measures using performance metrics like Normalized Discounted Cumulative Gain and Mean Average Precision. The results presented in this paper show that the TF-IDF scheme with Locality Sensitive Hashing Nearest Neighbor Algorithm using the Weighted Jaccard Distance can yield superior retrieval efficiency when contrasted with alternative approaches found in the existing literature.

Keywords: Document Retrieval; locality sensitive hashing nearest neighbor algorithm; weighted jaccard distance; term frequency - inverse document frequency

INTRODUCTION

The process of matching a given user query against a collection of free-text documents is known as document retrieval. These documents could be of any type, although they would primarily consist of unstructured text, including news stories, data collections, books, journals, surveys, statistics, paragraphs in a manual etc.

The emerging requirement for document retrieval in various fields of business, science, technology, education, entertainment, art and research leads to the use of various computer-based document retrieval methods and searching techniques. Search engines and document retrieval management tools are available online to support web-based document search. Basically, a document retrieval is getting the required document to the right people based on their search. Tremendous achievement in computer-based document retrieval has given way for innovative tools and techniques using the latest technology trends in searching. However, tools that supports in search and retrieval of the document should ensure the document's security and appropriate access to it (Asemi, Ko & Nowkarizi 2020).

A well sustained document retrieval should be able to search for documents based on keywords as well as other attributes such as date and author. The keyword choices are made based on the title and textual content of the documents and they have been indexed on all relevant fields. The usage of apt keywords by the user is decisive for an efficacious document retrieval with applicability to the required document. When the users lack acquaintance

on the on the search area, it makes the user to struggle in achieving the efficient retrieval. This leads to search results that is deficient to cover the searched topic ("Enhanced web document retrieval using automatic query expansion" 2023).

Early days document retrieval was based on Information Retrieval (IR) systems that used Boolean system (AND, OR and NOT). As there were shortcomings in using Boolean systems such as absence of ranking concept and difficulty in framing the search request query for Information retrieval, the document retrieval schemes in the current trend uses ranking methods (Singhal 2001).

Document retrieval using ranking method use IR systems that rank the documents by assessing its relevance to the user query. Most of the IR systems which are ranking based consign numeric score to the documents and rank the documents by their score. Many models have been developed to implement the ranking process of the documents. The models that have more significance and commonly used are vector-based model, the probabilistic model and the inference network model (Järvelin & Kekäläinenm n.d.). Vector Space model is the most lucrative model to program words and documents into a vector (Jalilifard et al. 2020).

In vector space model the text is represented as a vector of terms. A term may refer to the words and phrases. A text with the term is given a non-zero value. The document is assigned the numeric score for every query. Since a query is also text-based, it can be represented as a vector (query vector). The vector model evaluates the similarity between the query vector and the document vector to assign a numerical score to a document based on the query (Ababneh et al. 2014). Probabilistic models use the Probability Ranking Principle (PRP) in which the rank is assigned to the documents based on the diminishing probability of their relevance to the given query. This model does the estimation of probability of relevance of a document for a query and which is the significant component of the model. The estimation technique varies from one probabilistic model to another (Santos, Macdonald & Ounis 2010).

The most commonly used information retrieval techniques are implemented with inference network model. This model conceptualizes document retrieval as an evidence-based cognitive process, where various pieces of evidence from the document and search query are integrated to determine the probability of the query content matching the document. The ranking of the document is done based on the weight assigned to the term. The inference network model has shown tremendous improvement in the retrieval performance. At the same time the costs of computations can be paralleled with that of the traditional retrieval models, which enable large-scale data retrieval ("Parallel Computing" 2023).

The traditional keyword-based document retrieval system are not accurate, hence there is a need of the domain based ontology systems document processing and document retrieval. This method provides a feasible and superior quality domain ontology model for the retrieval of documents (Yu 2019). Retrieving biomedical documents is mainly required for lot of medical diagnosis and research. The authors used rule-based and deep learning-based methods so as to retrieve the data from the datasets and the relevant publications. The authors also used the "retrieval plus re-ranking" strategy to find the relevant datasets, and rank them using standardized ranking models (Wei. 2017).

Efficient retrieval of contents from scientific documents is carried out in Math Information Retrieval system (MathIRs). This work indexes mathematical expressions using a mechanism based on substitution trees and retrieve the relent documents efficiently (Pathak et al. 2017). The authors proposed an integrated approach for document retrieval to support change impact analysis. The authors employed a bag-of-words approach, representing each class or method as a collection of words, and combined it with a neural network-based information retrieval (IR) technique to derive conceptual coupling so as to improve the F-score measure (Wang et al. 2018).

The authors suggest retrieving scientific documents based on the topic of interest. They have extracted a set of IT key-phrases using the key-phrase extraction method. The outcome of the research was evaluated two approaches: the query-based approach involves retrieving articles from WoS based on specific search terms or keywords and the research area-based approach utilizes WoS categorizations and to identify relevant articles within a particular domain (Mohebi, Sedighi & Zargaran 2016). Nearest neighborhood algorithm is used for relevant document search as it is extensively used machine learning algorithms which is adaptive and simple (Uddin et al. 2022).

The remaining sections of this paper are organized as follows: Section 2 delves into the proposed work. Section 3 shows the implementation of Document Retrieval task; Section 4 includes the evaluation of the results. Finally, Section 5 discusses the conclusion and proposes future work directions.

PROPOSED WORK

The dataset consists of around 60K Wikipedia pages about the famous personalities. The main focus of this research work is to find the closest personalities of a given personality as a search key term. The overall process is grouped into data preparation, application the TF-IDF

algorithm to find the most significant terms of the documents and using the Nearest Neighbor search to find the relevant documents using various similarity measures. The proposed overall work is shown in flow diagram given in Figure 1.

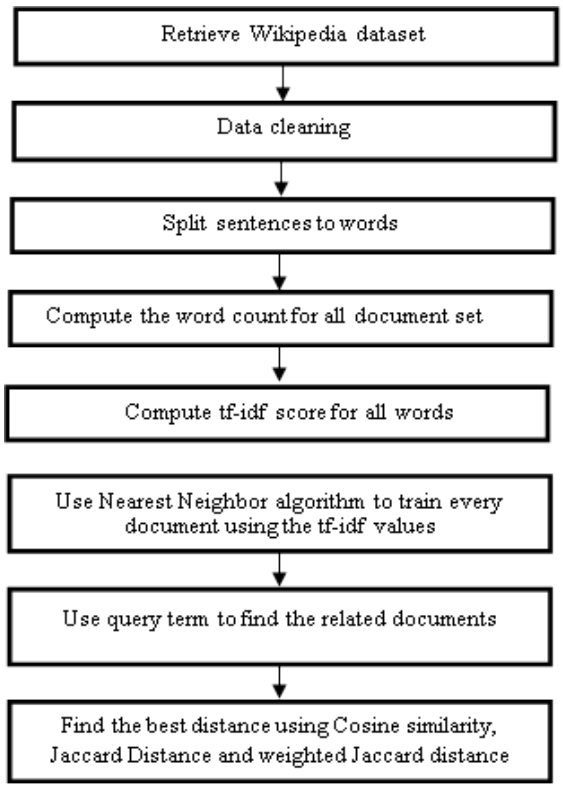

FIGURE 1. Overall Proposed Document Retrieval Process

TF - IDF ALGORITHM

Term frequency-inverse document frequency is represented by TF-IDF. It is the most commonly used scheme for assigning weight to the terms in information retrieval process which can be classified under inference network model in Information retrieval. The methodology of this algorithm is that to determine the relative frequency of the occurrence of a word in a document or group of documents with the inverse proportion of the word occurrences in the corpus. Words whose occurrence is rare has more TF-IDF number where as the words that are very common in all the documents such as preposition and articles (example: "is", "the' etc.) will have a lower value for TF-IDF weightage (Ponomarev 2022). Stop list - list of words that do not possess the ability to discriminate from one document to another. While indexing the document or formulation of query from the user search words, the words from the stop list are not included ("Tf-idf :: A Single-Page Tutorial - Information Retrieval and Text Mining" 2023).

NEAREST NEIGHBOR SEARCH

Nearest Neighbor Search (NNS) is one of the basic approach in machine learning, databases, signal processing, and a variety of other disciplines. When a NNS approach is applied to a database it returns the nearest records using some similarity measure (Cayton & Dasgupta 2007) (Weber et al. 1998).

The search of required data from the datasets are very hard due to heterogeneous data, varied conditions and system environment. Hence it is essential to create a technique that provides the best results. To address the challenges of NNS approach, various techniques with different methodologies have been developed. NNS search can be broadly categorized into two main components: the first involves selecting an appropriate data structure for indexing and maintaining data points, while the second focuses on employing a suitable algorithm to identify the nearest Neighbors of a given query point x (Abbasifard, Ghahremani & Naderi 2014).

COSINE SIMILARITY

The cosine similarity is a common and effective similarity measure employed in information retrieval and clustering tasks (Larsen & Aone 1999). The cosine similarity between two documents is a technique for comparing their similarity by calculating the cosine of the angle between their vector representations. The comparison between the documents are carried out based on the orientation (i.e.; the angle between the documents are taken and not the magnitude of each word count of each document). Cosine similarity measure is autonomous of document length. Cosine similarity measure will produce a metric that states how related are the documents by referring to the angles. When the angles are in the same direction then cosine similarity value is 1, when the angles are in the opposite direction then cosine similarity value is -1, when the angles are nearly orthogonal then cosine similarity value is 0 (Lakshmi & Baskar 2018).

JACCARD DISTANCE

The Jaccard index is a widely used measure for comparing the similarity, dissimilarity, and distance between datasets. It is calculated by dividing the number of common features between two datasets by the total number of features in both datasets. The Jaccard distance, on the other hand, is a measure of dissimilarity that is inversely related to the Jaccard similarity. It is calculated by subtracting the Jaccard similarity from 1 or, equivalently, by dividing the difference between the total number of features in both datasets and

the number of common features by the total number of features (Chahal 2016).

WEIGHTED JACCARD DISTANCE

Weighted Jaccard distance is applied to calculate the likeness between two sets using Jaccard but moderate the results using the relative frequency of each item within the assigned two sets. The Jaccard index determines the similarity between two sets by taking the ratio of the size of their intersection to the size of their union ("Weighted Jaccard Similarity" 2023).

IMPLEMENTATION

DATA PREPARATION

The primary step in retrieving similar documents is the data preparation. The dataset should be pre-processed in order to get the cleanest and formatted the data. Pre-processing transforms the raw text data into a well-defined etymological word set. Data pre-processing involves removing special characters and numbers, converting text to a consistent case format, tokenizing the text into individual words, eliminating stop words, and stemming.

The first step in the pre-processing stage is the removal of the special symbols since it changes the word count. Therefore, commas, dots, exclamation marks, semicolons, apostrophes and other special symbols were removed from the data set. Once the special symbols were removed then all the numerals were removed except the Roman numerals. After that, all strings were converted into the lowercase form so as to provide the uniformity in the text. Once these preliminary pre-processing steps were over, then the whole document content should be tokenized. Tokenization is the process of splitting the whole document into separate tokens or words, which will be used for the further analysis. Tokenization is followed by elimination the stop words, which does not give any relevance or meaning to the tokens. The last phase of the pre-processing process is stemming, which remove suffixes or prefixes from a token in order to find the "root word" or stem of a given word. Stemming was performed using the most popular algorithm called porter2 which is available with the python packages.

COMPUTING TF-IDF

The cleaned tokens were taken for further to find out the TF–IDF score for every token throughout the dataset. First the word frequency or the term frequency (TF) for every single word in each document was calculated, which gives the idea of the about the number of times every word occurred in the document. Term frequency can be found in many ways, such as the raw term frequency, logarithmically scaled frequency and the most commonly used is the augmented frequency. In this work most of the documents contain almost same number of words so it's not relevant to find the logarithm scale or map the frequency based on the size of the documents.

The word count list given in figure 2. shows the sample output of the word count operation for a single document and list each and every unique word and its occurrence.

```
: 1L, 'continued': 1L, 'presidential': 2L, 'husen': 1L, 'californias': 1L, 'equality': 1L, 'prize': 1L, 'lost': 1L, 'stimul
us': 1L, 'january': 3L, 'university': 2L, 'rights': 1L, 'july': 1L, 'gun': 1L, 'troop': 1L, 'withdrawal': 1L, 'brk': 1L, 'r
eferred': 1L, 'affordable': 1L, 'attorney': 1L, 'senate': 3L, 'regained': 1L, 'national': 2L, 'creation': 1L, 'related': 1L
, 'hawaii': 1L, 'born': 2L, 'taught': 1L, 'election': 3L, 'close': 1L, 'operation': 1L, 'insurance': 1L, 'sandy': 1L, 'afgh
anistan': 2L, 'initiatives': 1L, 'reform': 1L, 'house': 2L, 'review': 1L, 'representatives': 2L, 'current': 1L, 'state': 1L
, 'won': 1L, 'limit': 1L, 'victory': 1L, 'unsuccessfully': 1L, 'reauthorization': 1L, 'keynote': 1L, 'full': 1L, 'patient':
1L, 'august': 1L, 'degree': 1L, 'bm': 1L, 'mitt': 1L, 'attention': 1L, 'delegates': 1L, 'lgbt': 1L, 'job': 1L, 'harvard': 2
L, 'term': 3L, 'served': 2L, 'november': 2L, 'debt': 1L, 'care': 1L, 'received': 1L, 'great': 1L, 'libya': 1L, 'receive': 1
L, 'months': 1L, 'urged': 1L, 'foreign': 2L, 'american': 3L, 'protection': 2L, 'economic': 1L, 'act': 8L, 'military': 4L, '
```

FIGURE 2. Sample word count for a document

Once the term frequency is calculated, the next step is to find the inverse document frequency. The inverse document frequency of a word indicates whether the term is commonly used or infrequently used across all documents in a corpus. It is calculated by taking the logarithm of the ratio of the total number of documents in the corpus (N) to the number of documents containing the search term (n)

IDF(term, Documents) = LOG (N / n)

The TF-IDF score for every term in every document is determined by multiplying the corresponding TF and IDF values. The following table 1 shows the TF – IDF values for a single document with the higher values.

TABLE 1. TF – IDF values for terms in a document

Word	TF-IDF
khan	30.96
dabangg	19.20
kiya	18.39
biwi	18.08
kuch	18.08
pyar	17.18
hain	16.08
actor	14.79
salman	14.27
hum	14.23

REMOVING THE TERMS WITH LOW TF-IDF VALUE

The higher the TF – IDF value indicates that the term is more significant than the other terms, so it is required to filter out the terms with the lower TF – IDF value which does not contribute in the further process and also to reduce the size of the features. The TF – IDF values for all the documents are not in the same range, so it is required to find out a threshold value to filter out all the insignificant terms. For every document, the computed TF_IDF values will be grouped in 4 quartiles and the lower three quartiles will be skipped because those values are not much significant. The terms with the TF – IDF values greater than or equal to the upper hinge of the box plot were selected for the training process. Figure Figure 3(a), shows the box plot with four quartiles constructed for the TF – IDF values of a sample document. The upper hinge value of this box plot is 7.14, so all the terms with the TF – IDF values greater or equal to 7.14 were extracted as shown in Figure 3(b).

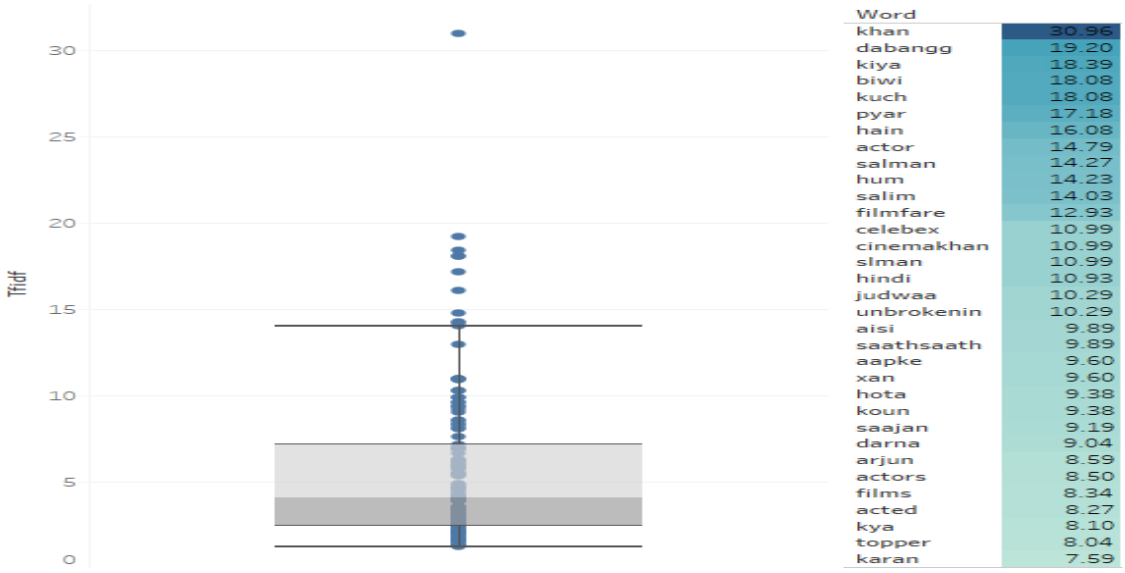

FIGURE 3(a). Box plot for TF – IDF values of a single document b). Significant terms

CREATING MODEL USING NEAREST NEIGHBORS ALGORITHM

Once the feature set with the TF – IDF values are prepared the next step is to create a nearest Neighbor model that will learn from the significant terms based on their TF – IDF values and to predict the document title. The Locality Sensitive Hashing (LSH) algorithm is used to find the nearest neighbors of the given search query. The proposed approach is implemented by creating three separate models using LSH algorithm for three different distant measures such as Jaccard, Weighted Jaccard and the Cosine distance with same set of significant features (Arya et al. 1998).

The created model was tested by providing the search key term, which will display the first 10 similar documents including the search document. The following figures Figure 4, 5, and 6 shows the search results of the nearest Neighbor search model for the search term “Salman Khan”, with different distance measures. The value shown in the figure is the distance between the search term and the

retrieved document. The search results were ranked based on the distance, if the distance is less indicating that the retrieved document is very close to the search term.

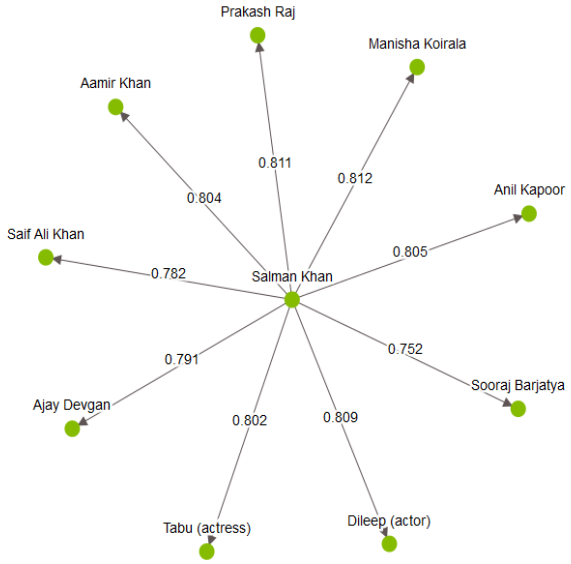

FIGURE 4. Top 10 documents retrieved using Jaccard distance

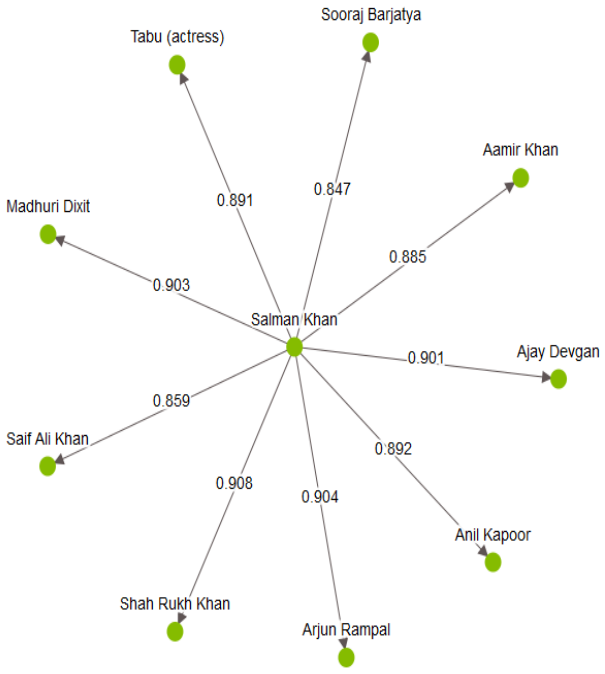

FIGURE 5. Top 10 documents retrieved using Weighted Jaccard distance

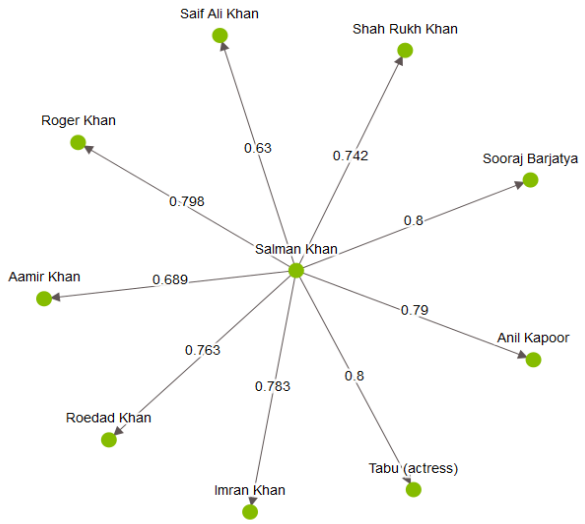

FIGURE 6. Top 10 documents retrieved using cosine distance

PERFORMANCE EVALUATION

The LSH based nearest Neighbor algorithm to find the most relevant documents from a set of documents using a search term was implemented in python and the results were analysed. The performance of the nearest Neighbor algorithm was analysed for 3 different distance parameters such as cosine distance, Jaccard distance and weighted Jaccard distance for 10 random query searches. Every search query returns the top 10 results sorted according to the similarity with the search keyword. The results will be ranked manually in the grade of 0 to 3 in the order of importance, 0 means the resulting document is not relevant, 3 means highly relevant, 1 or 2 less relevant and relevant respectively. From these grades and results the following parameters are calculated, the Gain (G), Cumulative Gain (CG), Discounted Cumulative Gain (DCG), Ideal Discounted Cumulative Gain (IDCG) and the Normalized Discounted Cumulative Gain (NDCG).

The NDCG values calculated for the three different distances are plotted as a line graph shown in Figure 7.

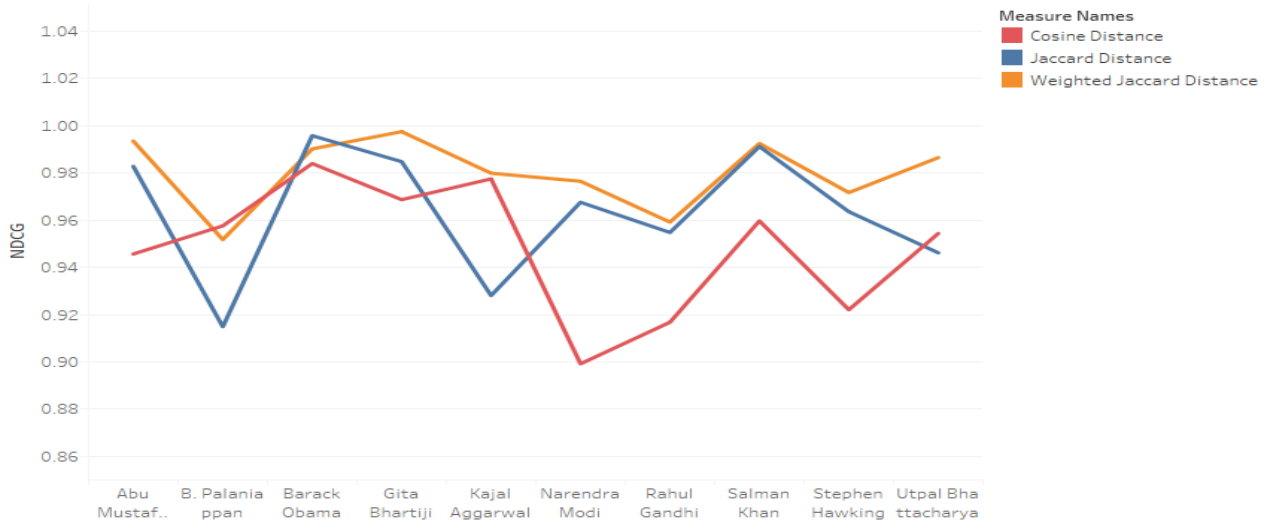

FIGURE 7. Normalized Discounted Cumulative Gain (NDCG) values for 10 search results

The graph shows the NDCG values for the 10 sample search keys for the selected distance measures. From the graph the Weighted Jaccard Distance has the higher value of NDCG which indicates that this distance measure is best suited for the document retrieval.

The average NDCG value for the Jaccard distance is 0.96298, weighted Jaccard distance is 0.97993 and cosine distance is 0.94857. The average NDCG values also indicate that the Nearest Neighbor search with the weighted Jaccard distance performs well in retrieving documents.

The next measure to check the performance of the algorithms is the Mean Average Precision (MEP), which shows the mean of the average precision of each search. The following graph Figure 8 shows that the weighted Jaccard distance has the highest value of MEP with 0.9941, followed by Jaccard distance with value 0.9726. This clearly indicates that Nearest Neighbor search with the weighted Jaccard distance outperforms the other distance measures.

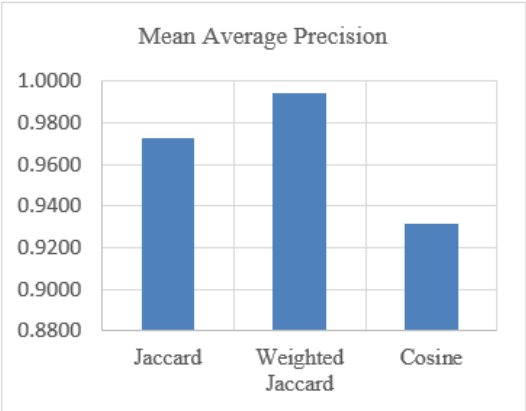

FIGURE 8. Mean Average Precision (MEP) for the different distance measures

CONCLUSION

The proposed method for retrieving documents related to different personalities extracted from Wikipedia demonstrates competitive performance compared to existing methods in the Personality Identification domain. The combination of Locality Sensitive Hashing Nearest Neighbor algorithm with Weighted Jaccard Distance and TF-IDF provides enhanced precision in document similarity measurement, enabling the retrieval of more relevant documents. Additionally, the introduction of a document centroid normalization technique further improves the effectiveness of information retrieval by enabling better discrimination between documents. The

algorithm is applied with different distance measures like Cosine, Jaccard and weighted Jaccard. Based on the performance evaluation, the Weighted Jaccard method provides high accuracy in retrieving documents. These findings suggest that the proposed method is a promising approach for retrieving personality-related documents from large corpora. The future work for this research involves enhancing the proposed method by integrating machine learning and deep learning algorithms for improved accuracy and adaptability (Pyingkodi et al. 2021), (Pyingkodi et al. 2023), (Varghese et al. 2023). The investigation of multimodal analysis and real-time retrieval represents key directions for expanding the method's versatility across diverse domains.

ACKNOWLEDGEMENT

The authors would like to thank University of Technology and Applied Sciences for their support.

DECLARATION OF COMPETING INTEREST

None

REFERENCES

Ababneh, J., Almomani, O., Hadi, W., El-Omari, N.K.T. & Al-Ibrahim, A. 2014. Vector space models to classify Arabic text. *International Journal of Computer Trends and Technology* 7(4).

Abbasifard, M.R., Ghahremani, B. & Naderi, H. 2014. A Survey on nearest neighbor search methods. *International Journal of Computer Applications* 95(25).

Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R. & Wu, A.Y. 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM* 45(6).

Asemi, A., Ko, A. & Nowkarizi, M. 2020. Intelligent libraries: A review on expert systems, artificial intelligence, and robot. *Library Hi Tech* 39(2): 412–434.

Cayton, L. & Dasgupta, S. 2007. A learning framework for nearest neighbor search. *Advances in Neural Information Processing System*

Chahal, M. 2016. Information retrieval using jaccard similarity coefficient. *International Journal of Computer Trends and Technology* 36(3).

Enhanced web document retrieval using automatic query expansion. 2023. https://onlinelibrary.wiley.com/doi/10.1002/asi.10341 [21 October 2023].

Jalilifard, A., Caridá, V.F., Mansano, A.F., Cristo, R.S. & Da Fonseca, F.P. 2020. Semantic sensitive TF-IDF to determine word relevance in documents. *International Journal of Scientific Research* 4(8).

Järvelin, K. & Kekäläinenm, J. n.d. IR evaluation methods for retrieving highly relevant documents. *SIGIR Forum* 51(2): 41–48.

Lakshmi, R. & Baskar, S. 2018. Efficient text document clustering with new similarity measures. *International Journal of Business Intelligence and Data Mining* 18(1).

Larsen, B. & Aone, C. 1999. Fast and effective text mining using linear-time document clustering. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Mohebi, A., Sedighi, M. & Zargaran, Z. 2016. Subject-based retrieval of scientific documents, case study: Retrieval of Information Technology scientific articles. *Library Review* 66(6/7): 549–569.,

Parallel Computing. 2023. http://research.microsoft.com/enus/um/people/gray/benchmarkhandbook/chapter8.pdf [20 October 2023].

Pathak, A., Pakray, P., Sarkar, S. & Das, D. 2017. MathIRs: Retrieval system for scientific documents. *Comp. y Sist* 21(2): 253–265.

Ponomarev, I. 2022. Development of an automated system for clustering text documents. *System technologie*s 1(138): 115–119.

Pyingkodi, M., N.R, W.B., Shanthi, S., Mahalakshmi, R. & Gowthami, M. 2021. Performance evaluation of machine learning algorithm for lung cancer. *International Journal of Aquatic Science* 12(3).

Pyingkodi, M., Thenmozhi, K., Karthikeyan, M., Chitra, K., Wilfred Blessing, N.R. & Kumar, S. 2023. Fruits quality detection using deep learning models: A meta- analysis. *5th International Conference on Inventive Research in Computing Applications*.

Santos, R., Macdonald, C. & Ounis, L. 2010. Exploiting query reformulations for web search result diversification. *Proceedings of the 19th International Conference on World Wide Web*

Singhal, A. 2001. Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin, Google press* 24(4).

Tf-idf :: A Single-Page Tutorial - Information Retrieval and Text Mining. 2023. https://tfidf.com/ [21 October 2023].

Uddin, S., Haque, I., Lu, H., Moni, M.A. & Gide, E. 2022. Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Sci Rep* 12(1).

Varghese, A., George, B., Sherimon, V., Shuaily, A. & Salim, H. 2023. Enhancing trust in alzheimer's disease classification using explainable artificial intelligence: Incorporating local post hoc explanations for a glass-box model. *Bahrain Medical Bulletin* 45(2).

Wang, W., He, Y., Li, T., Zhu, J. & Liu, J. 2018. An integrated model for information retrieval-based change impact analysis. *Scientific Programming* 1–13.

Weber, R., Schek, H.-J. & Blott, S. 1998. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces: *Proceedings of the 24rd International Conference on Very Large Data Bases*194–205.

Wei, W. 2017. Information Retrieval in Biomedical Research: From Articles to Datasets. *UC San Diego*.

Weighted Jaccard Similarity. 2023. https://mathoverflow.net/questions/123339/weighted-jaccard-similarity

Yu, B. 2019. Research on information retrieval model based on ontology. *Journal of Wireless Com Network* 1: 1–8.