Journal of Engineering Science and Technology
Vol. 19, No. 4 (2024) 1206 - 1220

DEVELOPMENT OF A MACHINE LEARNING MODEL FOR MULTIPLE MEMBERSHIP PUBLICATION DOCUMENTS

APRIANI P. PURFINI[1]*,
STEPHANIE B. R. HERSIANIE[2], RIO YUNANTO[1]

[1]Faculty of Computer Science, Universitas Komputer Indonesia, Bandung, Indonesia
[2]Faculty of Engineering, Universitas Wiralodra, Indramayu, Indonesia
*Corresponding Author: apriani.puti.purfini@email.unikom.ac.id

Abstract

This research aims to develop a learning model in the text mining process by modifying TFIDF. The methodology employed in this study consists of a literature review, data collection, data preprocessing, and document processing. Text mining is conducted using the title and keyword attributes in publication documents. The outcome of this research is a learning model with modified TFIDF to determine multiple memberships in publication documents. The modification of TFIDF weighting can group words according to their document field categories. In contrast, previous TFIDF methods could only group words per document, making it difficult to determine the field categories of publication documents. The results of this research are expected to provide a solution to achieve more accurate outcomes with the same data scheme. The modification of TFIDF weighting can group words according to their document field categories. These field categories consist of a collection of words from many documents that have characteristics in specific field categories. Longer documents tend to have higher TFIDF weights because they can contain more words. This condition can affect the comparison between long and short documents. TFIDF also does not consider the context surrounding words, so in some cases, context is crucial for understanding the meaning of words or phrases.

Keywords: Multiple membership, Publication document, TFIDF.

1. Introduction

Text Mining or Knowledge Discovery from Text (KDT) was first [1], referring to the process of extracting high-quality information from textual data [2]. One form of text mining is information retrieval systems. One stage in text mining is the pre-processing stage of documents or text. The main objective of this stage is to obtain key terms from text documents and enhance the relevance of words in text documents with categories. Document pre-processing is a process that plays a crucial role in text mining [3].

The limitation of information within a community drives individuals within a community to become members of another community (multiple membership). Previous research has identified many memberships in social networks [4] and proposed supervised variants of the TFIDF scheme. The intuition behind the essential IDF factor is that terms that frequently appear across the collection have little discriminatory power. Proposing a term weighting scheme that leverages the availability of past search results, consisting of queries containing specific terms, documents retrieved, and relevance assessments [5]. A document can have more than one document with similar category percentages (multiple categories), thus requiring a threshold value to determine when a document is declared multiple categories.

Determining multiple memberships of scientific publication documents faces difficulties when using traditional TFIDF methods because each word in traditional TFIDF is weighted per document. In contrast, the data scheme used requires per-word weighting per field category. These field categories are generated from attribute words in the title and keyword of scientific publication documents according to their scientific fields. This set of attribute words in the title and keywords is built into a learning model. Therefore, this research proposes developing a learning model by modifying TFIDF.

TFIDF is one method for document representation. Limitations within TFIDF include disregarding many details that ideally should be relevant when processing text, such as document length, frequency distribution, and others [6]. Therefore, to address its shortcomings, we propose 4 (four) vector representations to overcome some of the limitations of TFIDF. Another drawback of TFIDF is its consideration solely based on word occurrences. The more a word appears, the more significant it is deemed [7]. DF-ICF (Document Frequency Inverse Corpus Frequency) is employed to tackle TFIDF's limitation of only considering word occurrences. DF-ICF focuses more on the importance of a document, which is equally crucial to understanding a term or word [8]. TFIDF fails to represent words semantically. This drives [9] to propose SemTFIDF, which is TFIDF based on semantic similarity. Frequency count is not the sole distinguishing factor calculated in determining word weights[10].

Therefore, it is proposed that weight determination be accompanied by statistical data calculation on words. The utilization of LCS within TFIDF balances the occurrences of the same word sequences between the query and the text within the document. The presence of very long but irrelevant documents result in generated weights incapable of representing document relevance [11]. Modifying TFIDF with LCS can effectively enhance document retrieval [11].

2. Research Method

The research stages are illustrated in Fig. 1. The research begins with a literature review, which involves collecting and examining articles such as journals, proceedings, and books related to concepts, text mining methods, TFIDF, Vector Space Model, etc.

Fig. 1. Research stages.

Data collection to construct the model consists of scientific publication documents from each journal from the DBLP data source, categorized by the field of Information and Computing Science according to the Field of Research (FoR). Additionally, data sources include publication documents from the RPLP and Informatics KK publications. Data pre-processing involves tokenization, filtering, and other procedures to prepare data for model construction, including extracting titles and keywords and separating author names.

The document processing begins by dividing the data into training and testing data sourced from DBLP, RPLP, and Informatics KK publication documents. Subsequently, experiments are conducted to measure document similarity through supervised learning based on title attributes and keywords generated by the authors.

3. Results and Discussion

The model development process employs training data already labeled by category. This training data undergoes a learning process to generate vectors representing each category. The process comprises TFIDF weighting on each word from each category and the formation of category vectors, all done automatically.

3.1. Document preprocessing

The input for this stage consists of document attributes (titles and keywords) from scientific publication documents. These documents are sourced from DBLP and the author's scientific publication documents (publications from RPLP and Informatics KK). Document preprocessing is done to standardize formats, eliminate duplicate data, and check for inconsistent data. Document preprocessing involves case folding, tokenization, stop-word removal, and stemming. Stemming modification is conducted at this stage. The output of this stage is a list of keywords for both attributes that have undergone preprocessing.

Training and testing data are sourced from DBLP journals, with field category types conforming to the FoR classification standard. The data used in the model development process is the training data, which already has known category labels. Additionally, the author's publication documents from RPLP and Informatics KK are also used as training data, manually labeled beforehand. Meanwhile, the testing data is used in the process of measuring document query similarity against category vectors. Both training and testing data contain sets of title words and keywords

manually extracted from scientific publication documents. Both datasets have undergone document preprocessing, resulting in lists of title keywords and keywords.

The category types used in this model align with the FoR classification standard in the Information and Computing Sciences division, including AI and Image Processing (AI), Computational Theory and Mathematics (CT), Computer Software (CS), Data Format (DF), Distributed Computing (DC), Information System (ISM), Library and Information Science (LIS). The utilization of FoR in this research is because FoR provides clear sub-division of disciplines with consistent hierarchical levels within each disciplinary group. The training and testing data from DBLP journals are shown in the Table 1.

Table 1 shows the total training and testing data sourced from DBLP. Meanwhile, the publication documents from RPLP and Informatics KK from 2009-2013, used as training data, amount to 137 publications. The total number of authors to be analyzed for multiple memberships in their field is 26 individuals.

Table 1. Training and testing data from DBLP journals.

Type of Dataset	Total
Training Data (80% of the total data)	2,300
Testing Data (20% of the total data)	575
Total data	2,875

The preprocessing is a crucial process in the data mining stage [12]. The quality of data generated from preprocessing can affect the results of data mining processing. The preprocessing conducted on these documents is divided into two types: preprocessing related to text processing in document attributes and preprocessing related to author data. The document attributes used in text preprocessing are titles and keywords created by the authors. Document preprocessing in this study is carried out automatically. Meanwhile, the preprocessing of author data aims to eliminate duplicate author names, which generally have different spellings in various types of documents. Author data preprocessing is done manually.

Case folding aims to change all characters in the attributes to a single format (lowercase all). This format alignment is necessary because the writing of titles and keywords in scientific publication documents varies depending on the journal source. The case folding process is done by converting uppercase letters to lowercase. Only the letters 'a' to 'z' are accepted. Characters other than letters are considered delimiters so that they will be removed from the document, such as punctuation (&, #,:, -, ;) and numbers (0-9).

Tokenization is the process of breaking sentences into words (tokens). With this process, the input string appears more concise as it is displayed as individual words. The cutting of character sets in keywords is done based on comma characters. The problem encountered in tokenization is when a sentence uses a hyphen within it. For example, real-time, time-optimal. This word will have a different meaning if the combined words are tokenized into separate words. Therefore, to maintain the meaning of combined words containing punctuation marks within them, this case is addressed by combining the two tokens connected by punctuation into one token that has meaning, namely, real-time, time optimal.

Stop word removal is performed after the tokenization process. This process aims to eliminate frequently occurring words that are not used in processing. The removal of stop words aims to make the query search results more focused on important words that can represent a query [13]. For example, when a search engine searches with the query "how to develop information retrieval applications," the search engine will display more results related to the words "how" and "to" because these words are too common in English. Therefore, these terms can be ignored, allowing the search engine to focus more on retrieving pages containing the keywords "information," "retrieval," and "applications."

The purpose of stop word removal is to eliminate common words, resulting in relevant words according to the requested query in the matching process. These common words have word frequencies in several documents, thus not contributing to the context or information of the document. Stop-word removal generally follows existing stop-word lists, particularly in English [14]. A standard list of stop words for general text, including "a," "about," "able," "across," "after," "all," "almost," "also," and so forth. Building a stop words list can also be done by considering the IDF value of each word [15]. If a word has the smallest IDF weight, it means the word is spread across all documents and is common. This research also constructs an Informatics stop words list based on 7 (seven) categories used. The stop words list is manually constructed, considering IDF = 0. Here is some stop words related to the field of Informatics according to the data used in this study; system, image, model, data, network, algorithm, design, feature, pattern, program, group, software, and method.

The stemming process is performed after tokenization and stop word removal. This process aims to return words to their base form, increasing recall and providing relevant results in the word search process [16]. Considerations for applying stemming include: 1) Morphological variants of words that should have the same meaning must be mapped to the same stem. 2) Words with similar etymological forms but different meanings should not be mapped to the same stem. The stemming aims to reduce inflectional and derivative forms of a word to a common base form [17]. For example, the words "cars, car's, car" all have the same base form, "car." The main problem with stemming algorithms is obtaining the correct base form of a compound word. One issue encountered in the stemming process is over stemming and under stemming. Defined over stemming as excessive truncation of words, resulting in situations where different words with very different meanings produce the same stem [18]. For example, words like "generalized, general, generous, generating" all produce the stem "gener." Additionally, insufficient affix removal may lead to the same words not producing the same stem (under stemming).

To address the aforementioned issues, a modification will be made to the stemming process. This modification involves combining two stemming algorithms (hybrid stemming): a dictionary lookup table algorithm and a Porter stemming algorithm. A lookup table essentially stores root words in a table within a database along with their derived forms [19]. Words are manually entered into the table. The list of words entered into the database can reduce errors in over stemming and under stemming. When users input inflected words, the stemmer will search for the presence of these inflected words in the database. According to KBBI, inflection refers to a change in a word indicating various grammatical relationships.

Handling both cases involves creating a list of words entered into each rule table. This list is based on a word dictionary obtained from the website http://www.morewords.com. This website was chosen as a word dictionary option because it sources words from the Enhanced North American Benchmark Lexicon (ENABLE2K), which contains over 173,528 words. The list of words used, along with their derivative forms, is manually entered into tables within the database. The following sequence of processes outlines the steps taken to address stemming cases in this study.

3.2. Analysis of weighting with modified TFIDF

Weighting aims to determine the importance level of each word in a document. TFIDF is a method used to calculate the frequency of words most commonly used in information retrieval [20]. This method is also known for its efficiency, simplicity, and accurate results [21].

The TFIDF weighting matrix is presented with columns consisting of terms or words in each document (sentence), and the column represents the frequency count of words in the first document and so forth [22]. The documents referred to are sequences of sentences consisting of a collection of terms. Table 2 shows the weighting matrix.

Table 2. TF weighting.

term	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	14	0	17

TFIDF weighting in the model development process aims to construct category vectors. Therefore, the TFIDF weighting matrix commonly used undergoes modifications in its column structure. Table 2 is an example of a word weight matrix used in constructing category vectors. The matrix columns show a list of words derived from titles or keywords, IDF values, the frequency count of words in the first category (term frequency), the weight (w) of words towards the first category, and so forth.

The modification of the TFIDF word weighting matrix in titles can be observed in Table 3. The title word column contains terms or words from the titles that have undergone preprocessing. Furthermore, the IDF column contains the IDF values for each word row. These IDF values are obtained from $\log(\frac{D}{df_1})$, where D is the total number of categories, i.e., seven field categories, and df_1 is the number of categories that have a specific word.

Table 3. TFIDF weighting.

Word in Title	IDF	Number of word frequencies in Category 1	Word weights for Category 1	Number of word frequencies in Category n	Word weights for Category n
Word 1	0.845	1	0.845	0	0
Word 2	0	9	0	5	0
Word 3	0.544	2	1.088	1	0.544

Next, columns labeled as category 1, 2, 3...n contain the frequency count of a specific word possessed by a certain category. Subsequently, the column representing the weight of words towards a particular category indicates the weight of a word regarding its term frequency in a specific field category. This weight is obtained by multiplying the term frequency in a field with its IDF value (W = TF * IDF).

The difference between the modified TFIDF weighting matrix and the standard weighting matrix lies in the calculation of the frequency count of word occurrences. The modified TFIDF weighting calculates the frequency count of word occurrences based on the total words appearing in the document collection within each category.

3.3. Analysis of category vector length calculation

After each word in every category has been weighted, the next process is the calculation of the vector category length. The length of the vector category can be calculated using the following calculation:

$$|D_i| = \sqrt{\sum_{i=1}^{n} Di^2} \quad (1)$$

The length of the df_i category vector (Di) is obtained from the square root of the total squared weights of words in each category.

3.4. Analysis of the classification process

The classification process aims to measure the similarity between document queries from test data against the constructed category vectors. This process shares the same steps as the model-building process, which involves TFIDF weighting. However, TFIDF weighting is applied to the submitted query. Additionally, this process includes a sub-process for measuring the similarity between document queries and category vectors. The similarity measurement process is conducted after the document query has been vectorized. This process is carried out automatically.

TFIDF weighting marks the beginning of the similarity measurement between document queries and category vectors. The sequence of steps in this similarity measurement process consists of the following: It starts with searching for keywords identical to those in the desired query within the weight matrix. The submitted query typically includes a title or keywords from a scholarly publication document. The second step involves calculating the length of the query vector. The calculation of the query vector's length is obtained.

$$| Q | : \sqrt{\sum_{i=1}^{n} Qi^2} \quad (2)$$

The length of the query vector (Q) is obtained from the square root of the total squared term weights in the query. The third process involves calculating the dot product between the query and each category. The dot product calculation entails the total sum of the multiplication between the term weight values in the query and the term weight values found in the category vector field.

$$\mathrm{Sim}(\mathrm{Q},\mathrm{D}) = \mathrm{Cos}(\mathrm{Q},\mathrm{D}) : (\mathrm{Q} * \mathrm{D}))/((|\mathrm{Q}| * |\mathrm{D}|)\,) \quad (3)$$

The calculation of cosine similarity, as stated by Mandala and Setiawan [15], declares that the ranking process of a document is considered the selection process of document vectors close to the query vector. A higher cosine value indicates the

document is closer to a particular query. Several factors can influence the process of measuring document similarity, including a) IDF weight, the weight (w) of words towards categories, and vector length remaining constant. The addition or removal of categories can result in changes in these weights. b) The IDF value is influenced by word distribution. If a word is present in multiple categories, it will result in a smaller IDF value than a word in only one category. c) The higher the frequency of word occurrence, the higher the weight (w) of the word. IDF serves as a multiplier for word frequency (tf). d) Generally, the most negligible IDF weight can be used as a reference for stop word removal. If a word has the most negligible IDF weight, it means the word is distributed across all categories, resulting in a small IDF value.

The smallest IDF weight has a value of 0. IDF weight is obtained from the IDF formula, which is $\log(\frac{D}{df_i})$ For example, this thesis employs 7 field categories considered as the value ofW D. Then, the value of df_i is the type of category that has words distributed across all categories. If all categories have the same type of word, df_i will be 7. Therefore, the calculation of the smallest IDF weight in this study is $\log(\frac{7}{7})$ which equals 0.

It can be concluded that words with an IDF weight of 0 are words with the smallest IDF and are distributed across all categories. That words that appear frequently in various documents can be considered common terms and therefore are not significant [23].

It can be concluded that words with an IDF weight of 0 are words with the smallest IDF and are distributed across all categories. That words that frequently appear in various documents can be considered common terms and therefore their values are not significant [23].

3.5. Analysis of document similarity measurement results

Ensemble voting involves several steps: (1) Training multiple CNN models, (2) Generating predictions, (3) Conducting a vote, (4) Aggregating predictions, and (5) Determining the final prediction. This study utilized five models, each trained for 15 epochs, with a learning rate of 0.001.

All documents have been assessed for their similarity to category vectors in the community type determination stage. The outcome of this process is the cosine similarity value between the document query and each category vector. The higher the cosine similarity value of a category vector, the more similar the document is to that category.

The category vector type in the top rank has the highest cosine similarity value, hence this category vector type is referred to as the primary field. Meanwhile, the category vector type in the second rank is termed as the secondary field. Each document may have the possibility of similarity to two types of category vectors.

Figure 2 illustrates that Document A has a cosine similarity value of 0.228 with the CS category vector and a cosine similarity value of 0.011 with the ISM category vector. Document A has the highest cosine similarity value with the CS category, indicating that CS is referred to as the primary field in Document A. This can be corroborated by the vector space model in Fig. 3.

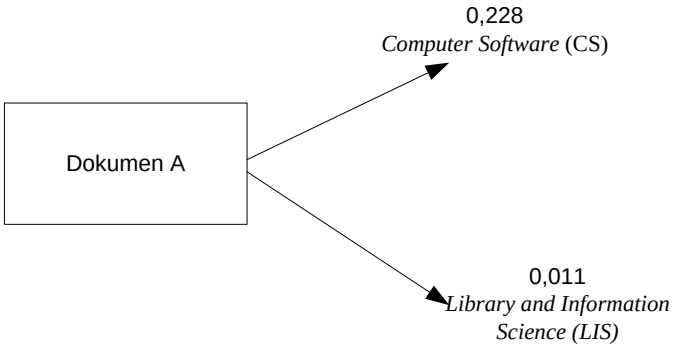

Fig. 2. Document a has achieved a cosine similarity value.

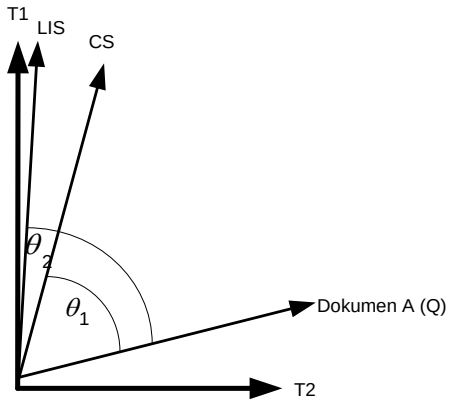

Fig. 3. Query of document a in vector space.

Figure 3 indicates that Document A bears a closer resemblance to the CS category vector compared to the LIS category vector. This is attributed to the angle formed between the query document vector and the CS category vector being smaller than that with the LIS category vector. The phenomenon observed in the similarity measurement process between query documents and category vectors is the variance in the ranking of cosine similarity values. Measuring the similarity between query documents and title attributes can yield rankings from first to second place for a document. Conversely, when measuring the similarity between query documents and keyword attributes, not all documents can secure first or second-place rankings.

Figure 4 illustrates the similarity measurement of Document A queries using title and keyword attributes. Similarity measurement between query documents and title attributes can yield two types of categories. In contrast, not all similarity measurements between query documents and keyword attributes can produce two types of categories. This is because keywords tend to contain more unique terms, meaning that keywords crafted by authors can vary, resulting in a document having only a limited range of category variations.

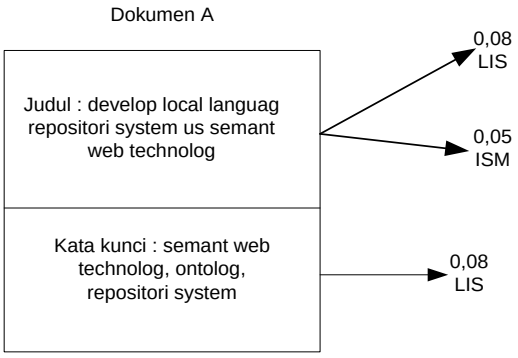

Fig. 4. Types of categories generated from the title and keywords.

3.6. Classification stage

The Classification stage aims to determine classes for unlabeled data. The input for this stage is preprocessed test data. The test data is used as query documents to measure their similarity to category vectors. The processes involved in this stage include TFIDF weighting on the query documents from the test data, calculation of query vector lengths, and calculation of cosine similarity between query documents and category vectors. The calculation of cosine similarity is carried out to determine the relevance of a category to the query, which is perceived as a similarity measure process [23]. The cosine similarity values obtained from this process are then ranked. The highest cosine similarity value indicates the increasing similarity of the query to that category. This cosine similarity calculation process produces cosine similarity values for the primary field (first rank) and secondary fields (second rank).

The TFIDF weighting can be modified by grouping words according to their document field categories. These field categories consist of a collection of words from many documents that have characteristics in specific field categories. The TFIDF value increases proportionally with how many times a word appears in a document but is balanced by the word's frequency in the corpus, which helps control the fact that some words are generally more common than others [24].

Multiple membership documents refer to situations where a document or scholarly work is presented or published by several team members or different authors. This situation is common in the academic and research world, especially in interdisciplinary and international collaborations [25, 26]. In many cases, a document or scholarly work will list all authors or team members as contributors to the work, which is an ethical practice to acknowledge each individual's contribution to research or collaborative projects. In multiple membership documents involving contributions from multiple authors or team members, it is important to treat each contributor fairly and acknowledge their contributions according to ethical practices and applicable publishing guidelines [27, 28].

The Term Frequency-Inverse Document Frequency (TFIDF) technique, including multiple publication documents, is commonly used in document classification. Document classification can utilize the TFIDF representation of each document as features to be used by classification algorithms such as k-Nearest Neighbors (k-NN), Naive Bayes, or Support Vector Machines (SVM) [29-31]. TFIDF can also be combined with pre-trained transformer models for document classification cases. Transformers, such as BERT (Bidirectional Encoder Representations from Transformers) or GPT (Generative Pre-trained Transformer), can be effective in understanding text and handling tasks like document classification [32].

Combining TFIDF with transformer models can utilize representations from both methods as features in the classification model [33]. The model can utilize TFIDF representations for frequently occurring words and representations from the transformer model for more complex contexts, then combine these features into a more complex classification model, such as neural networks or ensemble models [34, 35]. The ability of TFIDF to highlight important words in documents and the transformer model's ability to understand context and relationships between words can be experimented with and adjusted to ensure the effectiveness of this approach in document classification cases [36].

Some advantages of using TFIDF include its ability to highlight words that frequently appear in specific documents but rarely appear across the entire document collection. This ability aids in identifying important words and characteristics of the document [37, 38]. Common words like "the," "is," and "and" usually appear in many documents in the dataset; by using IDF, the weights of these common words are reduced, thus not dominating the TFIDF calculation. The ability to compare documents as TFIDF vectors enables easy comparison between documents by calculating the cosine similarity between these vectors. This is beneficial in document classification and information retrieval [39, 40].

Although TFIDF is a popular and useful method in text analysis and document classification, it has some limitations. It only considers the frequency and presence of words in documents without considering the semantic meaning of these words. In some cases, words with similar semantic meanings may have different TFIDF weights, even though they are similar in certain contexts [41, 42]. Longer documents tend to have higher TFIDF weights because they naturally contain more words. This condition can affect the comparison between long and short documents. TFIDF also does not consider the context surrounding words, so in some cases, context is crucial for understanding the meaning of words or phrases. The results of TFIDF are highly influenced by text preprocessing, such as tokenization, stop word removal, and stemming. If preprocessing is not done properly, the TFIDF results may not be optimal [43, 44].

4. Conclusion

A publication document may belong to more than one field of study. Determining the multiple memberships of publication documents using traditional TFIDF weighting encounters difficulties because this weighting method only groups words per document, making it challenging to determine the field categories of the publication documents. Modifying TFIDF weighting can group words according to their document field categories.

These field categories consist of a collection of words from many documents that exhibit characteristics in specific field categories. In many cases, a document will list all authors as contributors to the work, treat each author fairly, and acknowledge each author's contribution following applicable publishing guidelines. Longer documents tend to have higher TFIDF weights because they can contain more words.

This condition can affect the comparison between long and short documents. TFIDF also does not consider the context surrounding words, so in some cases, context is crucial for understanding the meaning of words or phrases.

References

1. Feldman, R.; and Dagan, I. (1995). Knowledge discovery in textual databases (KDT). P*roceedings of the First International Conference on Knowledge Discovery and Data Mining* (KDD'95), 95(1), 112-117.
2. Chen, M.S.; Han, J.; and Yu, P.S. (1996). Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and data Engineering*, 8(6), 866-883.
3. Ramasubramanian, C.; and Ramya, R. (2013). Effective pre-processing activities in text mining using improved porter's stemming algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(12), 4536-4538.
4. Kannan, S.; Gurusamy, V.; Vijayarani, S.; Ilamathi, J.; Nithya, M.; Kannan, S.; and Gurusamy, V. (2014). Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks*, 5(1), 7-16.
5. Okkalioglu, M. (2023). TF-IGM revisited: Imbalance text classification with relative imbalance ratio. *Expert Systems with Applications*, 217(1), 119578-119592.
6. Song, S.K.; and Myaeng, S.H. (2012). A novel term weighting scheme based on discrimination power obtained from past retrieval results. *Information processing & management*, 48(5), 919-930.
7. Chen, K.; Zhang, Z.; Long, J.; and Zhang, H. (2016). Turning from TF-IDF to TF-IGM for term weighting in text classification. *Expert Systems with Applications*, 66(1), 245-260.
8. Santhanakumar, M.; Columbus, C.C.; and Jayapriya, K. (2018). Multi term-based co-term frequency method for term weighting in information retrieval. *International Journal of Business Information Systems*, 28(1), 79-94.
9. Li, X.; Liu, N.; Yao, C.; and Fan, F. (2017). Text similarity measurement with semantic analysis. *International Journal of Innovative Computing, Information and Control*, 13(5), 1693-1708.
10. Hiemstra, D. (2000). A probabilistic justification for using tf× idf term weighting in information retrieval. *International Journal on Digital Libraries*, 3(1), 131-139.
11. Saadah, M.N.; Atmagi, R.W.; Rahayu, D.S.; and Arifin, A.Z. (2013). Information retrieval of text document with weighting TF-IDF and LCS. *Jurnal Ilmu Komputer dan Informasi*, 6(1), 34-37.

12. Tamilselvi, J.J.; and Gifta, C.B. (2011). Handling duplicate data in data warehouse for data mining. *International Journal of Computer Applications*, 15(4), 7-15.
13. Amarasinghe, K.; Manic, M.; and Hruska, R. (2015). Optimal stop word selection for text mining in critical infrastructure domain. *Proceedings of the* 2015 *Resilience Week* (*RWS*), Philadelphia, USA, 1-6.
14. Popova, S.; Kovriguina, L.; Mouromtsev, D.; and Khodyrev, I. (2013). Stop-words in keyphrase extraction problem. *Proceedings of the* 2013 14th *Conference of Open Innovations Association* (*FRUCT*), Espoo, Finland, 113-121.
15. Allioui, Y.; and El Beqqali, O. (2010). Personalized access towards a mobile neuroscience. *International Journal of Computer Science Issues* (*IJCSI*), 7(6), 169-181.
16. Joshi, A.; Thomas, N.; and Dabhade, M. (2016). Modified porter stemming algorithm. *International Journal of Computer Science and Information Technologies*, 7(1), 266-269.
17. Ceri, S.; Bozzon, A.; Brambilla, M.; Della Valle, E.; Fraternali, P.; Quarteroni, S.; and Quarteroni, S. (2013). An introduction to information retrieval. *Web Information Retrieval*, 18(8), 3-11.
18. Jabbar, A.; Iqbal, S.; Tamimy, M.I.; Hussain, S.; and Akhunzada, A. (2020). Empirical evaluation and study of text stemming algorithms. *Artificial Intelligence Review*, 53(1), 5559-5588.
19. Giri, V. (2021). MTStemmer: A multilevel stemmer for effective word pre-processing in Marathi. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(2), 1885-1894.
20. Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5), 503-520.
21. Qaiser, S.; and Ali, R. (2018). Text mining: Use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications*, 181(1), 25-29.
22. Hofmann, K.; Li, L.; and Radlinski, F. (2016). Online evaluation for information retrieval. *Foundations and Trends® in Information Retrieval*, 10(1), 1-17.
23. Hamzah, A. (2009). Temu kembali informasi berbasis kluster untuk sistem temu kembali informasi teks bahasa indonesia. *Jurnal Teknologi*, 2(1), 1-7.
24. Rehman, A.; Aslam, N.; Abid, K.; and Fuzail, M. (2023). The impact of COVID-19 on E-Learning: Context-based sentiment analysis discourse using text mining. *VAWKUM Transactions on Computer Sciences*, 11(1), 184-203.
25. Yaacob, W.F.W.; Nasir, S.A.M., Yaacob, W.F.W., and Sobri, N.M. (2019). Supervised data mining approach for predicting student performance. *Indonesian Journal of Electrical Engineering and Computer Scienc*, 16(3), 1584-1592.
26. Alami, N.; Meknassi, M.; En-nahnahi, N.; El Adlouni, Y.; and Ammor, O. (2021). Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling. *Expert Systems with Applications*, 172(1), 114652-114677.

27. Al-Thwaib, E.; Hammo, B.H.; and Yagi, S. (2020). An academic Arabic corpus for plagiarism detection: Design, construction and experimentation. *International Journal of Educational Technology in Higher Education*, 17(1), 1-26.
28. Jain, D.; Borah, M.D.; and Biswas, A. (2024). A sentence is known by the company it keeps: Improving legal document summarization using deep clustering. *Artificial Intelligence and Law*, 32(1), 165-200.
29. Jaiswal, M.; and Das, S. (2021). Detecting spam e-mails using stop word TF-IDF and stemming algorithm with Naïve Bayes classifier on the multicore GPU. *International Journal of Electrical & Computer Engineering*, 11(4), 2088-8708.
30. Perwira, R.I.; Yuwono, B.; Siswoyo, R.I.P.; Liantoni, F.; and Himawan, H. (2022). Effect of information gain on document classification using k-nearest neighbor. *Register*, 8(1), 50-57.
31. Aulia, M.A.I.; and Kurniawati, Y.E. (2022). Pengembangan model klasifikasi dokumen artikel teks berita olahraga dan bukan olahraga dalam bahasa indonesia menggunakan algoritma support vector machine. *KALBISIANA Jurnal Sains, Bisnis dan Teknologi*, 8(2), 2279-2291.
32. Weng, M.H.; Wu, S.; and Dyer, M. (2022). Identification and visualization of key topics in scientific publications with transformer-based language models and document clustering methods. *Applied Sciences*, 12(21), 11220-11230.
33. Gomes, L.; Da-Silva, T.R., and Côrtes, M.L. (2023). BERT-and TF-IDF-based feature extraction for long-lived bug prediction in FLOSS: A comparative study. *Information and Software Technology*, 160(1), 107217-107227.
34. Saifullah, S.; Dreżewski, R.; Dwiyanto, F.A.; Aribowo, A.S.; Fauziah, Y.; and Cahyana, N.H. (2024). Automated text annotation using a semi-supervised approach with meta vectorizer and machine learning algorithms for hate speech detection. *Applied Sciences*, 14(3), 1078-1088.
35. Suhirman, S.; Rianto, R.; Santosa, I.; and Yunanto, R. (2023). The ensemble method and scheduled learning rate to improve accuracy in CNN using SGD optimizer. *Journal of Engineering Science and Technology*, 18(6), 2779-2792.
36. Yunanto, R.; Wibowo, E.P.; and Rianto, R. (2023). A bert model to detect provocative hoax. *Journal of Engineering Science and Technology*, 18(5), 2281-2297.
37. Bounabi, M.; Elmoutaouakil, K.; and Satori, K. (2021). A new neutrosophic TF-IDF term weighting for text mining tasks: Text classification use case. *International Journal of Web Information Systems*, 17(3), 229-249.
38. Lubis, A.R.; Nasution, M.K.; Sitompul, O.S.; and Zamzami, E.M. (2021). The effect of the TF-IDF algorithm in times series in forecasting word on social media. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(2), 976-996.
39. Hasugian, P.M.; Manurung, J.; Logaraz, L.; and Ram, U. (2021). Implementation of TF-IDF and cosine similarity algorithms for classification of documents based on abstract scientific journals. *Infokum*, 9(2), 518-526.
40. Yunanda, G.; Nurjanah, D.; and Meliana, S. (2022). Recommendation system from microsoft news data using TF-IDF and cosine similarity methods. *Building of Informatics, Technology and Science (BITS)*, 4(1), 277-284.

41. Adebiyi, A.A.; Ogunleye, O.M.; Adebiyi, M.; and Okesola, J.O. (2019). A comparative analysis of TF-IDF, lsi and lda in semantic information retrieval approach for paper-reviewer assignment. *Journal of Engineering and Applied Sciences*, 14(10), 3378-3382.
42. Lan, F. (2022). Research on text similarity measurement hybrid algorithm with term semantic information and TF-IDF method. *Advances in Multimedia*, 2022(1), 1-11.
43. Kalra, V.; Kashyap, I.; and Kaur, H. (2022). Improving document classification using domain-specific vocabulary: Hybridization of deep learning approach with TFIDF. *International Journal of Information Technology*, 14(5), 2451-2457.
44. Kumar, A.; and Garg, G. (2023). Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets. *Journal of ambient intelligence and humanized computing*, 14(5), 5327-5342.