International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 3
ISSN: 2088-5334

Medical Record Document Search with TF-IDF and Vector Space Model (VSM)

Lukman Heryawan [a,*], Dian Novitaningrum [b], Kartika Rizqi Nastiti [b], Salsabila Nurulfarah Mahmudah [b]

[a] *Department of Computer Science and Electronics, Universitas Gadjah Mada, Sekip Utara Bulaksumur, Yogyakarta 55281, Indonesia*
[b] *Master Program in Computer Science, Universitas Gadjah Mada, Sekip Utara Bulaksumur, Yogyakarta 55281, Indonesia*
Corresponding author: [*]lukmanh@ugm.ac.id

***Abstract*—The growth of medical record documents is increasing over time, and the various types of diseases and therapies needed are increasing. However, this has not been followed by an effective and efficient search process. This study aims to deal with search problems that often take a long time with search results that are not necessarily as expected by building a search model for medical record documents using the vector space model (VSM) and TF-IDF methods. The VSM method allows retrieval of results that are not the same as the search queries entered by the user but are expected to provide still results relevant to the user's desired needs. The model development process was taken based on the data in the FS_ANAMNESA and FS_DIAGNOSA columns, followed by preprocessing, which consists of deleting blank lines, lowercase, removing punctuation marks, HTML tags, stop words, excess spaces between words, and normalizing typo words, then forming a TF-IDF matrix based on the frequency of occurrence of each word feature, and followed by the calculation of the similarity value of the search query compared to medical record documents based on the cosine similarity formula. The retrieval results were all columns of each existing medical record document and were sorted based on 10 rows with the highest similarity value. The model evaluation results were based on 1000 medical record documents and tested with 20 search queries in this study, which gave an average precision value of 0.548 and an average recall value of 0.796.**

***Keywords*— Medical records; preprocessing; cosine similarity; TF-IDF; evaluation metric.**

Manuscript received 17 Oct. 2023; revised 9 May 2024; accepted 19 May. 2024. Date of publication 30 Jun. 2024.

I. INTRODUCTION

Medical record data is kept for at least 25 years from the date of the last visit [1]. One indicator of the-quality-of-health facility services is speed and accuracy in providing medical records [2]. According to the Regulation of the Indonesian Ministry of Health number 129/Menkes/SK/II/2008 it explains that one of the minimum standards in a service at a health facility is to be able to provide medical records within 10 minutes and have information completeness of 100% [3]. But the fact that in a hospital in Indonesia is still 65.75%, it takes > 10 minutes for the patient's medical record data to be successfully displayed, and the accuracy level is 87.5% [4].

In line with the Medical Practice Act in the elucidation of Article-46-paragraph (1), what-medical-records-mean are files that contain patient notes and documentation identity, examination, treatment, activities, and other services provided to patients [5]. The definition of medical records is strengthened through Minister of Health Regulation (Permenkes) No. 269/2008, that the type of medical record data might be in the form of text (both organized and narrative), digital images (if digital radiology is used), or a combination of the two has been implemented), sound (for example, heart sounds), video, or in the form of bio-signal such as Electrocardiography (ECG) recordings [6].

Information retrieval (IR) in computer science is obtaining relevant information, large-scale electronic text, and other human language data sets that are being represented, searched for and otherwise manipulated in this scenario [7]. There are various categories of information retrieval parts; text operations-(including the selection of words in the query as well as the document), converting documents or queries into term indexes (index words),-query formulation (giving weight to query word index), and rating (look for document that are related to the query and rank them and return the document based on conformity with queries) [8]. Research and implementation of IR have been widely carried out in various fields, including searching for medical record documents. Faridah et al. [9] conducted a study to retrieve images from an extensive image archive based on image

query content in the medical field. Mutinda et al. [10] researched to capture the semantic similarity between Japanese clinical texts (Japanese clinical STS) by producing a publicly available Japanese dataset.

The Vector Space Model (VSM) is a mathematical paradigm for representing any object as a vector to measure the degree of similarity. The underlying idea behind this strategy is that words with similar meanings will have vectors that are closer to each other in this reduced space [11]. The Vector Space Model is an information retrieval system model that compares each query and document as an n-dimensional vector. Each dimension in the vector is represented by one term. The terms used are usually based on the terms in the query or keywords, so terms in the document but not in the query are typically ignored. The Vector Space Model is frequently utilized for relevance ranking, information retrieval, indexing, and filtering [12].

One of the most popular similarity measurements for VSM is Cosine similarity, and one way to encode textual documents into vectors is Term Frequency—inverse Document Frequency (TF-IDF). Wahyudi et al. [13] conducted research using cosine similarity with TF-IDF as a weighting scheme for searching JSON files relevant to a given query. Rofiqi et al. [14] conducted a study to find news documents based on similarities using TF-IDF.

The ranking includes stages with the help of cosine similarity. Cosine Similarity is a method that analyzes similarity, which has a function to get the desired word term based on the query and the distance to sort it [15], [16]When the results are more similar, the two objects being evaluated are said to be more similar. Several previous studies have discussed the classification of documents relating to cosine similarity. Ristanti et al. [17] conducted research using Similarity in Cosine, calculating the similarity between two things papers using this method. The benefit of this method is that it is not affected by document length but rather by the importance of the terms in each text and thus has a low error rate. Using this method's application's performance testing as a basis in comparison to 126 instances of economic article journals using the K-Fold Cross Validation approach, 6 folds for each data randomization set were tested 6 times, the average accuracy results were 57,79%, precision was 57,79%, and recall was 62.96%.

Precision is the ratio of the number of relevant documents found to the total number of documents found. Meanwhile, the second evaluation uses recall. Recall is the ratio of the number of-relevant-documents found to the overall number of documents in the collection considered relevant [18]. Another implementation of previous research is to search for similar themes in the synopsis of Indonesian novels using the General Vector Space Method. The dataset has more than 1500 records. Then, test data for 300 data were divided by 150 search test data and the rest for classification testing. So that the result is a recall of 90% and a precision of 85% [19].

In this study, the process of retrieving a collection of medical record documents from a dataset totaling 126,483 lines of data but only taking 1000 lines of data. The dataset is SOAP Medical Record data originating from REKMED, a cloud-based patient medical record recording application. This dataset has 5 columns containing `FS_ANAMNESA`, `FS_ACTIONS`, `FS_TERAPI`, `FS_PHYSICAL_NOTES`, and `FS_DIAGNOSIS`. The method used is to perform data preprocessing, calculate similarity with cosine similarity, rank based on cosine similarity, and use several evaluation matrices. The study's findings should be able to provide a search model for medical record documents that match the queries entered by users.

II. Materials and Method

The following is a description of the methodology used in this study. Fig. 1 shows the preprocessing process and Fig. 2 shows the similarity calculation process.

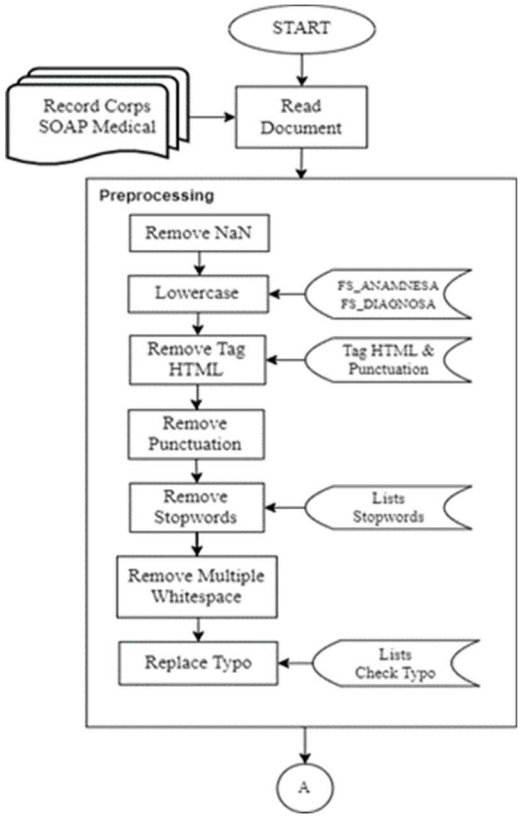


Fig. 1 Flowchart of the preprocessing stage

A. Preprocessing

Preprocessing is the first step in cleaning and normalizing existing medical record data. This step aims to extract a list of terms from the data source that will add significant value to the user's original query. Medical record data contains 5 columns, namely `FS_ANAMNESA`, `FS_ACTIONS`, `FS_THERAPY`, `FS_PHYSICAL_NOTES`, and `FS_DIAGNOSIS`. In this study, text data was taken from two columns, namely `FS_ANAMNESA` and `FS_DIAGNOSA`, with a total of 1000 rows of data. The following are details of each preprocessing stage carried out in this study:

1) Removes blank rows: Rows without data in either the `FS_ANAMNESA` or `FS_DIAGNOSA` columns are deleted for the next stage.

2) Lowercase: where each letter is converted to lowercase.

3) Removes punctuation and HTML tags: Medical record data has a lot of punctuation and HTML tags such as <p>,

, and so on, so it needs to be removed to make further processing more efficient.

4) Remove stop words: Words that have no significant meaning, such as conjunctions, prepositions, or other words, are deleted so that there is not too much noise.

5) Remove excess spaces between words or characters.

6) Normalization: it was done by replacing typo words: such as 'peerut' to 'stomach', 'jln' to 'road', 'riwy' to 'history', and so on. This study collected 1290 pairs of words used as references to replace words in medical record texts.

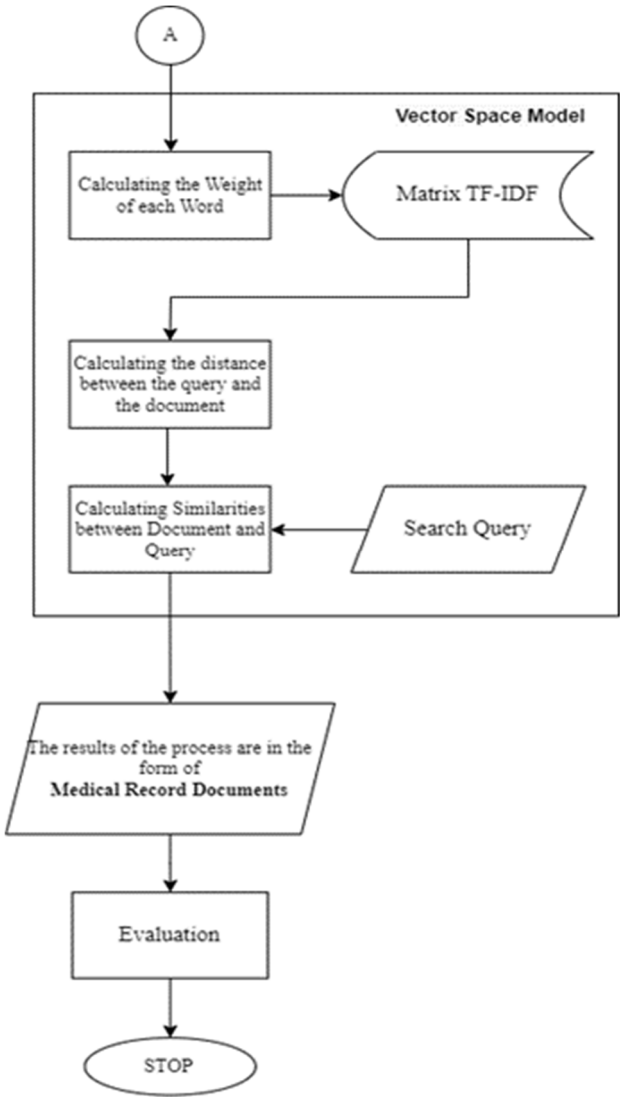


Fig. 2 Flowchart of similarity calculation

B. Vector Space Models (VSM)

The Vector Space Model is a vector that is used to replace the document model by utilizing a representation of the text approach in each word and is used in a spatial approach [20]. Vectors are formed from all the terms or words that make up the document. The term is a collection of words consisting of several different meanings [21], [22]The vector in each element represents the weight of words that have a relationship between the document and the query. The cosine angle is used to find the value of the two vectors for each document and query weight.

Meanwhile, the formula to calculate using the Cosine Similarity technique the closeness of values between-two-documents between the vectors in the documents and the vectors in the query [23]. The Vector Space Model works with tokenization, especially the phase of cutting each word in the input string that composes it and breaking the document, which was converted into a word frequency table. The table is a vector and can be stored as an array. All the words in the document are formed into one, which is called a term. There is a vector representation of each document, which is compared to the terms that have been constructed [24].

The process of calculating vector values in the Vector Space Model (VSM) requires three stages, namely:

1) Calculate each word's weight with Term Frequency - Inverse Document Frequency (TF-IDF): Term Frequency - Inverse Document Frequency (TF-IDF). TF-IDF is a way to calculate word weight based on the frequency of its occurrence in a corpus. The TF-IDF is a method of determining the importance of words within texts or corpora as numerical data [25]. The TF-IDF algorithm was used to evaluate the importance of words in the textual corpus [26]. The frequency of words is represented by TF, which indicates the number of times they appear in the corpus of Equation (1). The weight of each word is calculated from the occurrence of the word and compared to the total number of words in the corpus:

$$tf(t,d) = \frac{f_d(t)}{max\, f_d(w)} \quad (1)$$

$$idf(t,D) \ = ln\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right) \quad (2)$$

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \quad (3)$$

where $f_d(t)$ is the frequency of occurrence of the word t in document D and D is document corpus. IDF shows the level of importance of a word in the corpus as a whole. The IDF value is obtained by calculating the inverse logarithm value of the-proportion-of-documents containing specific words in Equation (2). The proportion value is taken from the total number of documents in the corpus divided by the number of documents containing a particular word. The logarithmic value of this proportion is the IDF value. The TF-IDF weight is calculated by multiplying the two values, as shown in Equation (3). The greater the TF-IDF value, the more important the word in question is in the corpus.

2) Calculating the distance between the query and the document. To calculate the distance between the query and the document, this study utilizes the `linalg.norm` function from the numpy library in Python. This function returns the norm value between vectors, which in this case represents the distance between two text documents. Calculations in the `linalg.norm` function is based on the Frobenius formula [27], also popularly known as the Euclidean distance. Equation (4) shows the formula of the Frobenius norm.

$$\left||A|\right|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{i,j}\right|^2} \quad (4)$$

where $\left||A|\right|_F$ = matrix norms and $a_{i,j}$ is the query and document weights

3) Calculation of Cosine Similarity: Cosine Similarity is a measure for calculating distances used in data as vectors from documents. Documents contain data that can consist of hundreds or even thousands of attributes. Each attribute represents a term or word that contains a value, meaning the frequency with which it occurs in a particular document. The vector in the document contains the frequency-of-the-number of times a word appears in a document. The calculation for the term value in each document is the similarity of the two vectors in the dimensional space obtained from the cosine at an angle of the multiplication between the two vectors being compared because the cosine at a value of 0 is 1 and less than 1 for the other angle. A cosine value of 0 indicates that the two vectors are orthogonal to each other and have no match. The higher the match between vectors, the smaller the angle and the greater the cosine value [28][29]. Two vectors are said to be similar when the value of their cosine similarity is 1. The calculation of cosine similarity is explained in Equation (5).

$$Sim(\alpha) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A1 \cdot B1}{\sqrt{\sum_{i=1}^{n}(A1)^2 \cdot \sum_{i=1}^{n}(B1)^2}} \quad (5)$$

where:
A = document vector
B = query vector
$A \,.\, B$ = product of vector A and vector B
$|A|$ = length of vector A
$|B|$ = length of vector B
$|A||B|$ = cross product between $|A|$ and $|B|$
α = the angle formed between vectors A and B

C. Evaluation

Precision and Recall is one of the most frequently performed evaluation tests on retrieval systems. Precision measures the accuracy of the number of documents that can be found and considered relevant by the search process for document search purposes. Meanwhile, recall is the ratio of the quantity of relevant documents. found to the overall number of records in the collection that are considered relevant [30]. Equation (6) describes precision, and Equation (7) describes recall.

$$Precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Number\ of\ documents\ retrieved} \quad (6)$$

$$Recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Number\ of\ relevant\ documents} \quad (7)$$

III. RESULTS AND DISCUSSION

To determine the performance of the developed information retrieval model, researchers used several included queries. Tables 1 and 2 list 10 queries used to test the model. The following is an example of a document that was successfully retrieved. Fig. 3 shows all the columns successfully retrieved while Fig. 4 shows the ranking, scores, document number, and FS_ANAMNESA columns more clearly. In Fig. 4, each successfully retrieved document has a weight obtained from calculating cosine similarity. The weight is used to determine the ranking of successfully retrieved documents. The higher the weight value of a document, the higher the ranking of the document. The higher the ranking of a document, the more relevant the document is to the query entered.

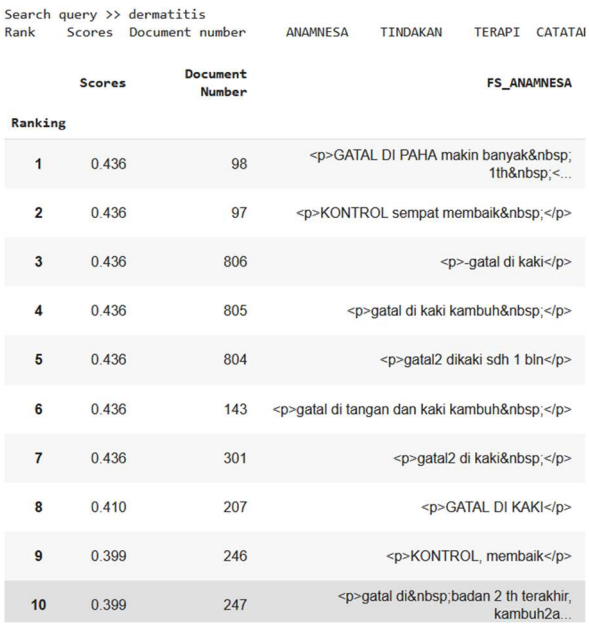
TABLE I
LIST QUERY FS_ANAMNESA COLUMN IN MODEL TESTING

FS_ANAMNESA		Number of Document
English	Indonesian	
control	control	10
toothache	gigi sakit	10
lung cancer	ca paru	10
right shoulder pain	nyeri bahu kanan	6
pain	nyeri	4
earache	telinga nyeri	4
right knee hurt	lutut kanan sakit	4
left knee pain	lutut kiri sakit	4
cough and cold	batuk pilek	2
cough	batuk	2

TABLE II
LIST QUERY FS_DIAGNOSA COLUMN IN MODEL TESTING

FS_DIAGNOSA		Number of Document
English	Indonesian	
pregnant	hamil	10
acute bronchitis	bronkitis akut	10
neurosa	neurosa	10
dermatitis	dermatitis	10
et	et	10
bph	bph	10
tooth abscess	abses gigi	10
meningioma	meningioma	10
kb	kb	4
rhinitis	rhinitis	4

Fig. 3 All columns that were successfully retrieved



Search query >> dermatitis

Rank	Scores	Document number	ANAMNESA	TINDAKAN	TERAPI	CATATAI

Ranking	Scores	Document Number	FS_ANAMNESA
1	0.436	98	GATAL DI PAHA makin banyak 1th <...
2	0.436	97	KONTROL sempat membaik
3	0.436	806	-gatal di kaki
4	0.436	805	gatal di kaki kambuh
5	0.436	804	gatal2 dikaki sdh 1 bln
6	0.436	143	gatal di tangan dan kaki kambuh
7	0.436	301	gatal2 di kaki
8	0.410	207	GATAL DI KAKI
9	0.399	246	KONTROL, membaik
10	0.399	247	gatal di badan 2 th terakhir, kambuh2a...

Fig. 4 Ranking, scores, document number, and FS_ANAMNESA columns that were successfully retrieved

Average precision and average recall are the evaluation matrices used in this study. Average precision determines the model's ability to obtain results that match the entered query, while average recall is used to obtain any document containing the entered query. From the queries that have been entered, average precision and average recall are obtained as shown in Table 3.

TABLE III
VALUE OF TEST RESULTS WITH PRECISION AND RECALL

Column	Average Precision	Average Recall
FS_ANAMNESA	0.522	0.816
FS_DIAGNOSA	0.574	0.781
Average	**0.548**	**0.796**

From the retrieve process that was carried out, the results were quite good, with an average-precision-of 54.8% and an average recall of 79.6%. The low precision and high recall results indicate that the model can predict more true positives.

TABLE IV
COMPUTATION TIME IN EACH CORPUS

Corpus	Computing Time (minutes)	Computing Time (seconds)
Without preprocessing	2.46778	8883.626732
Preprocessing	1.1561	4162.1

From the results in TABLE IV, the resulting computational time in the preprocessing corpus is indeed faster when compared to the corpus without preprocessing. However, if you look at the computational time required to process 1000 data, it takes 4162.1 seconds, then it can be concluded that running is very slow. Therefore, not all the datasets were used in this experiment, only 1000 data were used. So, the more time it takes to process the data, the heavier the computer's processor will work, thus affecting the bandwidth usage required to be greater. The experiment was again carried out using random data with 1000, 2000 and 3000 trials made, then the results were obtained.

TABLE V
RESULTS FROM FS_ANAMNESA COLUMN

	1000 random data	2000 random data	3000 random data
Average Precision	0.56	0.68	**0.78**
Average Recall	**0.83**	0.58	0.43

TABLE VI
RESULTS FROM FS_DIAGNOSA COLUMN

	1000 random data	2000 random data	3000 random data
Average Precision	0.63	0.75	**0.80**
Average Recall	**0.67**	0.50	0.32

Table 5 is the results of calculations using the FS_ANAMNESA column. TABLE VI is the calculation result using the FS_DIAGNOSA column. However, the results of the two are similar, namely the average precision value is better when using 3000 rows of random data, namely 0.78 for FS_ANAMNESA and 0.80 for FS_DIAGNOSA. Meanwhile, the average recall can get a value of 0.83 for FS_ANAMNESA and 0.67 for FS_DIAGNOSA.

Evaluation performance, which is still relatively low, can be caused by several factors, including (i) the data used is still small when compared to the total number of datasets; (ii) The preprocessing stage for RekMed data still needs to resolve the word typo. One of them can be seen from the FS_DIAGNOSA column, which explains a lot of medical words, so it requires the help of medical personnel to translate the meaning of these words; (iii) The dataset still contains various words that have not been normalized, such as writing abbreviations, writing dates that are not in the same format, and using + (positive) and - (negative) signs to describe a certain disease; (iv) Diversity in the use of language, such as words written in Indonesian, Javanese and English. In the future, it might be a suggestion for further research to improve the preprocessing stage of the dataset with the aim of equating the meaning of words into an agreed language so that users can understand what is written.

IV. CONCLUSION

The model evaluation results based on 1000 medical record documents and testing with 20 search queries in this study gave an average-precision-value of 0.548 and an-average recall value of 0.796. Low precision and high recall values indicate the model can predict more true positives. The retrieval results and evaluation values depend on the number of documents in the corpus and the queries entered, so many documents will likely be retrieved later, and the precision and recall values may change if there are adjustments to the number of documents and the use of other search queries. The search process in this study uses a cosine similarity calculation which only assesses the-similarity-between-documents-based-on-the closeness of the distance between the document and the query syntactically, while searching for RekMed documents requires consideration of semantic proximity, which other methods can implement.

ACKNOWLEDGMENT

This work was partially supported by the Department of Computer Science and Electronics, Universitas Gadjah Mada under the Publication Funding Year 2024.

REFERENCES

[1] I. Rosadi and M. I. Purnama, "Analysis Of Time Analysis Of Outstanding Medical Records To Improve The Quality Of Services At Dustira Hospital, Cimahi," KESANS: International Journal of Health and Science, 1(1), 1-5, 2021, doi: 10.54543/kesans.v1i1.2.

[2] "Peraturan Menteri Kesehatan Republik Indonesia nomor 24 Tahun 2022 tentang Rekam Medis."

[3] "Menteri Kesehatan Republik Indonesia Nomor 129/Menkes/SK/II/2008 tentang Standar Pelayanan Minimal Rumah Sakit."

[4] E. P. Widya Rita, R. Indrawati, and L. Widjaja, "A Service Quality Review of Medical Record Department In Private Hospital, South Jakarta," *Journal of Multidisciplinary Academic 101 JoMA*, vol. 05, no. 02, 2021.

[5] Indonesia, "Undang-Undang Nomor 29 Tahun 2004 tentang Praktik Kedokteran," Jakarta, 2004.

[6] Menteri Kesehatan Republik Indonesia, "Peraturan Menteri Kesehatan Republik Indonesia Nomor 269/MENKES/PER/III/2008 tentang Rekam Medis," *2008*.

[7] Stefan. Büttcher, C. L. A. Clarke, and G. V. Cormack, *Information retrieval : implementing and evaluating search engines*. MIT Press, 2010.

[8] K. Dalimunthe and B. H. Hayadi, "Information Text Retrieval untuk Pencarian Data Penilaian Mengacu pada Saran dari Pengunjung Menggunakan Vector Space Modelimplementasi." *Journal Computer Science and Information Technology (JCoInT)*, vol. 5, no.1, 2022.

[9] F. Faridah, K. Munadi, and F. Arnia, "Aplikasi Histogram Discrete Cosine Transform (DCT) untuk Sistem Temu Kembali Citra Termal Berbasis Konten," *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, vol. 2, no. 1, pp. 38–42, 2019.

[10] F. W. Mutinda, S. Yada, S. Wakamiya, and E. Aramaki, "Semantic Textual Similarity in Japanese Clinical Domain Texts Using BERT," *Methods Inf Med*, vol. 60, pp. E56–E64, Jun. 2021, doi: 10.1055/s-0041-1731390.

[11] S. Henry, C. Cuffy, and B. T. McInnes, "Vector Representations of Multi-Word Terms for Semantic Relatedness," *J Biomed Inform*, vol. 77, pp. 111–119, Jan. 2018, doi: 10.1016/j.jbi.2017.12.006.

[12] J. Yubo, D. Xing, W. Yi, and F. Hongdan, "A Document-Based Information Retrieval Model Vector Space," in *2011 Second International Conference on Networking and Distributed Computing*, IEEE, 2011, pp. 65–68. doi: 10.1109/ICNDC.2011.21.

[13] E. Wahyudi, S. Sfenrianto, M. J. Hakim, R. Subandi, O. R. Sulaeman, and R. Setiyawan, "Information Retrieval System for Searching JSON Files with Vector Space Model Method," in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, IEEE, 2019, pp. 260–265. doi:10.1109/ICAIIT.2019.8834457.

[14] M. A. Rofiqi, Abd. C. Fauzan, A. P. Agustin, and A. A. Saputra, "Implementasi Term-Frequency Inverse Document Frequency (TF-IDF) untuk Mencari Relevansi Dokumen Berdasarkan Query," *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol. 1, no. 2, pp. 58–64, Dec. 2019, doi: 10.28926/ilkomnika.v1i2.18.

[15] A. T. Adiyanto and D. Handayani, "Information Retrieval Sistem Kearsipan Pencarian Dokumen di Dinas Pemberdayaan Perempuan dan Perlindungan Anak Kota Semarang Menggunakan Metode Vector Space Model," *Jurnal Mahajana Informasi*, vol. 7, no. 1, 2022, doi: 10.51544/jurnalmi.v7i1.2538.

[16] R. Noor Santi, S. Eniyati, R. Retnowati, and H. Yulianton, "Penggunaan Sistem Temu Kembali Dalam Pencarian Kata Untuk Terjemahan Al Quran", Proceeding SENDI_U, pp. 247-252, Jul. 2019.

[17] P. Y. Ristanti, A. P. Wibawa, and U. Pujianto, "Cosine Similarity for Title and Abstract of Economic Journal Classification," in *2019 5th International Conference on Science in Information Technology (ICSITech)*, IEEE, 2019, pp. 123–127, doi:10.1109/ICSITech46713.2019.8987547.

[18] M. Tohir, D. Andariya Ningsih, N. Yuli Susanti, A. Umiyah, and L. Fitria, "Comparison of the Performance Results of C4.5 and Random Forest Algorithm in Data Mining to Predict Childbirth Process," 2023. doi: 10.21512/commit.v17i1.8236.

[19] Munif, M, E. Setyati and Y. Kristian, "Pencarian Tema Sejenis Sinopsis Novel Bahasa Indonesia Dengan Menggunakan GVSM", Joutica, vol. 6, no. 2, p. 492, Sep. 2021, doi: 10.30736/jti.v6i2.676.

[20] S. Harlina, R. D. Lillikwatil, K. Aryasa, C. Susanto, S. Sapriadi, and E. T. Alfriady, "Klasifikasi Sentimen Tweet Mengenai Covid-19 pada Twitter Di Indonesia Dengan Metode Vector Space Model," *CogITo Smart Journal*, vol. 8, no. 2, pp. 422–433, 2022, doi:10.31154/cogito.v8i2.405.422-433.

[21] E. Fitriani, R. E. Indrajit, *and* R. Aryanti, "Penerapan Model Information Retrieval untuk Pencarian Konten Pada Perpustakaan Digital," *Perspektif: Jurnal Ekonomi dan Manajemen Akademi Bina Sarana Informatika*, vol. 15, no. 2, pp. 170–176, 2017.

[22] O. Shahmirzadi, A. Lugowski, and K. *Younge*, "Text Similarity in Vector Space Models: a Comparative Study," in *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, IEEE, 2019, pp. 659–666. doi:10.1109/ICMLA.2019.00120.

[23] I. N. Wiyana, I. N. Purnama, and I. B. K. Sudiatmika, "Analisis Perbandingan Metode Vector Space Model dan Levenshtein Distance Dalam Sistem Temu Kembali Informasi pada Perpustakaan Digital STMIK Primakara (Primakara Library)", JUTIK, vol. 8, no. 4, Oct. 2022.

[24] K. Andesa, "Penerapan Metode Vector Space Model Pada Komunitas Jaringan Sosial (Studi Kasus Pada STMIK-AMIK Riau)," *Sains dan Teknologi Informasi*, vol. 1, no. 1, pp. 52–56, 2012.

[25] R. K. Ibrahim, S. R. M. Zeebaree, K. Jacksi, M. A. M. Sadeeq, H. M. Shukur, and A. Alkhayyat, "Clustering Document based Semantic Similarity System using TFIDF and K-Mean," in *2021 International Conference on Advanced Computer Applications (ACA)*, IEEE, 2021, pp. 28–33. doi: 10.1109/ACA52198.2021.9626822.

[26] M. Chiny, M. Chihab, O. Bencharef, and Y. Chihab, "Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms," Scitepress, May 2022, pp. 15–20. doi:10.5220/0010727500003101.

[27] G. H. Golub and C. F. Van Loan, "Matrix Computations, baltimore," *The John and Hopkins Press Ltd*, p. 81, 1996.

[28] B. S. Lancho-Barrantes and F. J. Cantu-Ortiz, "Quantifying the Publication Preferences of Leading Research Universities," *Scientometrics*, vol. 126, no. 3, pp. 2269–2310, Mar. 2021, doi:10.1007/s11192-020-03790-1.

[29] K. Orkphol and W. Yang, "Word Sense Disambiguation using Cosine Similarity Collaborates with Word2vec and WordNet," *Future Internet*, vol. 11, no. 5, p. 114, 2019, doi: 10.3390/fi11050114.

[30] D. P. P. Joby, "Expedient Information Retrieval System for Web Pages Using the Natural Language Modeling," *Journal of Artificial Intelligence and Capsule Networks*, vol. 2, no. 2, pp. 100–110, 2020, doi: 10.36548/jaicn.2020.2.003.