Available online at JDR Website: http://journal.unublitar.ac.id/jdr

Journal Of Development Research, 5 (1), May 2021, Pages 12-19

DOI: https://doi.org/10.28926/jdr.v5i1.143

Implementation of Weighted Tree Similarity and Cosine Sorensen-Dice Algorithms for Semantic Search in Document Repository Information System

Abdurrosyiid Amrullah[(1)], Indra Gita Anugrah[(2)]

Universitas Muhammadiyah Gresik, Indonesia

E-mail: [(1)]abdurrosyiid_170602@umg.ac.id, [(2)]indragitaanugrah@umg.ac.id

Received: 24 March 2020; Revised: 6 May 2021; Accepted: 15 May 2021

Abstract

Document search has several approaches, including full-text search, plain metadata search and semantic search. This study uses the Weighted Tree Similarity algorithm with the Cosine Sorensen-Dice algorithm to calculate the semantic search similarity. In this study, document metadata is represented in the form of a tree that has labeled nodes, labeled branches and weighted branches. The similarity calculation on the subtree edge label uses Cosine Sorensen-Dice, while the total similarity of a document uses the weighted tree similarity. The metadata structure of the document uses the taxonomy owner, description, title, disposition content and type. The result of this research is a document search application with taxonomic weight on file storage. From the experimental results combination Weighted Tree Similarity method with Tanimoto Cosine has an average recall of 58%, 88% precision, and 83% accuracy, while the combination of Weighted Tree Similarity with Cosine Sorensen-Dice has an average recall value of 66%, precision 88%. and accuracy 85%. Combination of Weighted Tree Similarity with Cosine Sorensen-Dice has better than the combination of Weigthed Tree Similarity with Tanimoto Cosine for search documents at the University of Muhammadiyah Gresik with a average recall value of 66% and an average accuracy of 85%. Similarity value on text labels using Cosine Sorensen-Dice is also influenced by the number of terms and documents in the repository.

Keywords: weighted tree similarity; semantic search; cosine similarity; sorensen dice similarity;

INTRODUCTION

As the number of documents we manage increases, the need for information retrieval becomes important. As more and more documents are stored, the search process becomes increasingly difficult. The usual approach to finding information from a document is usually a query. The use of queries in data retrieval is done by matching words, so that the search results will determine the presence or absence of words in the database. In contrast to information retrieval, information retrieval is an attempt to process data with the aim of obtaining a relationship from that data. The data in this case is a collection of documents containing words. To look for relationships between words, it is usually done in the textual analysis process (Anugrah & Sarno, 2017). This data connection is the main focus of information retrieval.

Each document usually has a structure, this structure will be used in the document search process using metadata search. Several document search algorithms use metadata, one of which is the Weighted Tree Similarity (WT Similarity) method. The WT Similarity algorithm generates a tree similarity value which is carried out by visiting the lowest node (leaf) and

then calculating the similarity of each branch pair (branch) through the weight of the edge that connects the leaf to the branch. The problem with WT Similarity is in the calculation of the similarity of pairs of nodes, where the effect of labels from the edges that connect between nodes. If the labels on the edges have the same meaning in the sense of being exactly the same (exact match) then it has a weight of 1 and otherwise if the labels on the edges are not exactly the same (non exact match) then the weight value is 0 (Basmalah Wicaksono et al., 2016).

Several related studies that underlie this research include research (Basmalah Wicaksono et al., 2016) In this study, It was concluded that the search by using the Weighted Tree Similarity Method gave better precision values compared to VSM even though the recall value was smaller than VSM, as evidenced by system and expert testing.

Meanwhile, (Alkaff et al., 2020) In this research use Weighted Tree Similarity and Content-Based Filtering, from the test results using five test scenarios, it was found that the system succeeded in providing good performance with a precision value of 88%.

From the research (Adi P & Palgunadi, 2014) results found that the combination of weighted tree similarity with tanimoto cosine resulted in a better search. by having a precision value of 100% and a recall of 84.44%.

Research (Putro & Thamrin, 2018) obtained the results of the similarity function assessment in sentence pairs from the three functions, dice similarity has the best similarity score to calculate sentence similarity, whereas euclidean distance has a poor similarity score for calculating sentence similarity.

From several previous studies it can be concluded that the weighted tree similarity method can be used in document search and in the study (Adi P & Palgunadi, 2014), the results of combining weighted tree similarity with tanimoto similarity have low recall results, namely 63.94%, while weighted tree similarity with cosine similarity is 80.89%. weighted tree similarity by combining tanimoto and cosine results in better searches by increasing recall to 84.44%. in research (Putro & Thamrin, 2018) the results of the comparison of cosine, dice and euclidean, dice similarity has the best similarity score to calculate the similarity of sentences. So that in this study a combination analysis of weighted tree similarity will be carried out by combining cosine similarity and sorensen-dice similarity, whether it can provide a better search than tanimoto cosine when used in the file search case at the University of Muhammadiyah Gresik

MATERIALS AND METHODS

The method used in the semantic search research used Weighted Tree Similarity (WT Similarity) to measure the similarity of tree structures and the use of the Vector Space Model (VSM) and cosine sorensen-dice to measure the semantic similarities between the edge labels being compared. Cosine sorensen-dice is a combination of cosine similarity with sorensen-dice similarity. The document data used is in accordance with the letter structure and document storage application at the University of Muhammadiyah Gresik (UMG). The storage structure for letters and documents consists of the title, owner, description, contents of the disposition and type or description of the letter.

At the initial stage, document extraction will be carried out according to the structure which is then represented in the tree structure. Subsequently, paired sub-tree similarities were measured to be compared. The similarity measurement of subtrees is done by taking every two vertices connected to one side (adjacent node). This paired contiguous node will be calculated

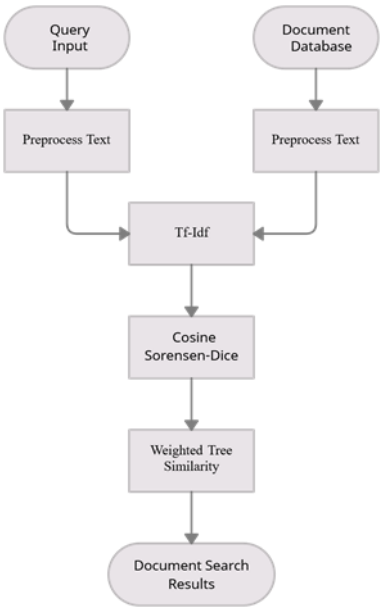

Figure 1 research overview

using a Weighted Tree Similarity. In the process of measuring the similarity of adjacent nodes in pairs, there is a process of measuring the similarity of edge labels by taking text labels to be carried out at the pre-processing stage of text, then look for the similarity of the text of the labels compared using cosine sorensen-dice. The text of the labels being compared. The next step, after knowing the similarities of the subtrees, is continued by looking for similarities to the trees being compared. Figure 1 shows an overview of the research methods used.

A tree consists of several subtrees, while each subtree has at least two nodes and one connected edge (Adi P & Palgunadi, 2014). The branch or branch itself is a subtree consisting of at least two connected nodes (adjacent node) and one of these nodes is a branch and a leaf.

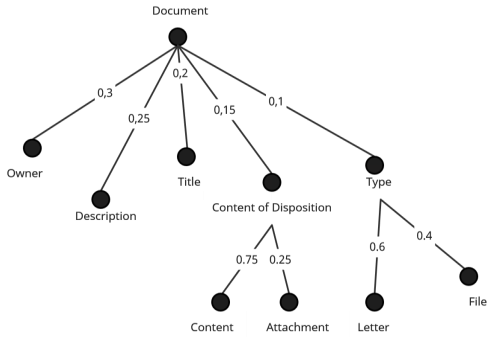

Figure 2 Example Tree

Figure 2 is an example of tree representation in the research carried out, namely the owner, description, title, content disposition, and type. This study emphasizes on the search results for the owner's sub-tree branch which has a weight of 0.3 compared to other branches of the subtree. Total of weights at the tree branch level is 1.

RESULT AND DISCUSSION

This chapter describes the results and discusses calculating the document's similarity to the input query. The following is the process of calculating the semantic search using a combination of the Weighted Tree Similarity method to calculate the similarity of the structure and the application of the cosine sorensen-dice used to calculate the semantic similarity of the subtree labels being compared.

Table 1 is an example of the document structure to be processed.

Table 1. Sample data

Doc	Structure	Sentence
D1	**Title**	undangan rapat evaluasi kerja dosen
	Description	undangan rapat kerja dosen
	Owner	Wakil Rektor 2
	Content of Disposition	baik mbak sani yth terima kasih
		Ada
	Type	Surat
		-

Preprocess Text

This process is performed on each edge label of the document tree and from the input query. Some of the steps taken at the text preprocess stage are as follows:

Case Folding

This process is used to normalize text, convert text to lowercase and remove punctuation marks.

Example:

Query

Undangan untuk rapat evaluasi kerja dosen ! .

Becomes

undangan untuk rapat evaluasi kerja dosen

Stopword Removal

After the case folding process is carried out, this stage is done by eliminating conjunctions or words that have no meaning or meaning.

Example:

Query

undangan untuk rapat evaluasi kerja dosen

Becomes

undangan rapat evaluasi kerja dosen

Stemming

The results of the stopword removal process still produce words that have affixes so that the stemming process must be carried out. The stemming process removes word affixes, either prefix,

insertion or suffix. This study use Sastrawi Indonesian stemmer.
Example:
Query

undangan rapat evaluasi kerja dosen

Becomes

undang rapat evaluasi kerja dosen

TF/IDF

After obtaining the preprocess text results, then TF / IDF calculations will be carried out. This stage is carried out to get the weight of a word and is carried out on each document subtree. For example, calculations will be performed on the title subtree.

Table 2 Example of TF / IDF Calculation

term	Q	d1	d2	d3	df	d/ df	idf
undang	1	1	1	0	2	1.5	0.176
rapat	1	1	1	0	2	1.5	0.176
evaluasi	1	1	0	0	1	3	0.477
kerja	1	1	1	0	2	1.5	0.176
dosen	1	1	1	0	2	1.5	0.176
informa-si	0	0	0	1	1	3	0.477
terima	0	0	0	1	1	3	0.477
bantu	0	0	0	1	1	3	0.477
ukt	0	0	0	1	1	3	0.477

information:
term: the word obtained after processing the previous text
Q: The number of terms contained in the query
d-n: the number of terms in the nth document
df: the number of terms in all documents
d: the number of all documents
idf: log (d / df)

Table 3 TF / IDF results

term	TF/IDF			
	Q	d1	d2	d3
undang	0.176	0.176	0.176	0
rapat	0.176	0.176	0.176	0
evaluasi	0.477	0.477	0	0
kerja	0.176	0.176	0.176	0
dosen	0.176	0.176	0.176	0
informasi	0	0	0	0.477
terima	0	0	0	0.477
bantu	0	0	0	0.477
ukt	0	0	0	0.477

The results in table 3 are obtained from the IDF results * the number of terms contained in the query and the sample document include terms "undang" in Q is 1 and idf term "undang" is 0.176 so the result is 1 * 0.176 = 0.176 and soon.

Similarity Calculations

The next process after obtaining the word weight from the TF/IDF process, calculating the similarity of the leaf node subtrees using the cosine sorensen-dice in Equations (1), (2), and (3).

$$Cosine = \frac{\sum(D1 \times D2)}{\sqrt{\sum(D1)^2+\sum(D2)^2}} \quad (1)$$

$$Sorensen\ Dice = \frac{2 \times \sum(D1 \times D2)}{\sum(D1)+\sum(D2)} \quad (2)$$

$$CS = Cosine \times Sorensen\ Dice \quad (3)$$

Information:
CS = cosine sorensen-dice
∑ = Total Data
D1 = The first sentence that will compare equations.
D2 = The second sentence, which will compare the similarities.

An example of data that will be calculated for the similarity is an input query with a leaf node subtree title from D1.

Query : undang rapat evaluasi kerja dosen
Title D1 : undang rapat evaluasi kerja dosen

Cosine Similarity is calculated using equation (1):

$$= \frac{4 \times (0.176 * 0.176) + (0.477 * 0.477)}{\sqrt{4 \times (0.176^2) + 0.477^2} \times \sqrt{4 \times (0.176^2) + 0.477^2}}$$

$$= \frac{0.351433}{\sqrt{0.351433} \times \sqrt{0.351433}} = \frac{0.351677}{0.351677} = 1$$

Meanwhile, Sorensen-Dice Similarity is calculated using equation (2):

$$= \frac{2 \times ((4 \times (0.176 * 0.176)) + (0.477 * 0.477))}{(4 \times (0.176^2) + 0.477^2) + (4 \times (0.176^2) + 0.477^2)}$$

$$= \frac{2 \times 0.351433}{0.351433 + 0.351433} = \frac{0.702866}{0.702866} = 1$$

Table 4. Similarity Calculation Results

Document	Title	Owner	Description	Contents of Disposition		Type	
				Content	Attachment	Letter	File
D1	1	0	0. 48	0	0	0	0
D2	0. 5897	0	0. 48	0	0	0	0
D4	0. 2649	0	0. 48	0	0	0	0
D5	0	0	0. 0145	0. 0487	0	0	0
D6	0. 0004	0	0. 0233	0. 0219	0	0	0
D7	0. 0212	0	0. 0065	0. 0124	0	0	0
D8	0. 0004	0	0. 0002	0. 0489	0	0	0
n…..	……	…..	…..	…..	……	…..	…..

After knowing the value of cosine similarity and sorensen-dice similarity, we will look for the value of cosine sorensen-dice using equation (3):

$$= 1 \times 1 = 1$$

After being implemented to all data the similarity will be calculated, the results are as shown in Table 4.

Weighted Tree Similarity

From the results of the calculation of the similarity of all subtrees of document tree leaf nodes, the total tree calculation will be carried out using the input query. To determine owner-branch similarity, the owner's node similarity is multiplied by the average weight of owner-branches in Figure 2, 0 * (0.3 * 0.3) / 2 yields owner-branch similarity. The algorithm then looks for similarities to the next branch node description, title, disposition content because this node is not a leaf, the algorithm will go down to calculate the similarity of the contents and attachment of the branch because there are no leaves, the results will be summed to determine the similarity value of the disposition content node, then the average weight of the disposition content branch to calculate the value similarity, proceed to type branch because this branch is also not a leaf, it will go down to the letter branch and type to calculate the similarity value. The following is an example of calculating the similarity of document tree D1 to Q using equation (4).

$$WTS = \sum\left(\frac{A(Si)(wi+wi')}{2}\right) \quad (4)$$

Information :
A (Si) = similarity in leaf nodes
wi = weighted tree arc weighted pair
wi '= weighted tree arc weighted pair

$$0 \times \left(\frac{0.3 \times 0.3}{2}\right) + 0.48 \times \left(\frac{0.25 \times 0.25}{2}\right) + 1 \times \left(\frac{0.2 \times 0.2}{2}\right) + \left(0 \times \left(\frac{0.75 \times 0.75}{2}\right) + 0 \times \left(\frac{0.25 \times 0.25}{2}\right)\right) \times \left(\frac{0.15 \times 0.15}{2}\right) + \left(0 \times \left(\frac{0.6 \times 0.6}{2}\right) + 0 \times \left(\frac{0.4 \times 0.4}{2}\right)\right) \times \left(\frac{0.25 \times 0.25}{2}\right) = 0.32$$

The result of the query equation with document d1 is 0.32. After the calculation of the weighted tree similarity algorithm is implemented on all data, the results are shown in Table 5, the higher the document similarity value, the document order is in the top position.

Implementation System

Implementation system of weighted tree similarity and cosine sorensen-dice on file search and system interfaces:

Table 5 Calculation Results of Weighted Tree Similarity

Document	WTS
D1	0.32
D2	0.234
D4	0.173
D5	0.0091
D6	0.0084
D7	0.0073
D8	0.0056
n….	……

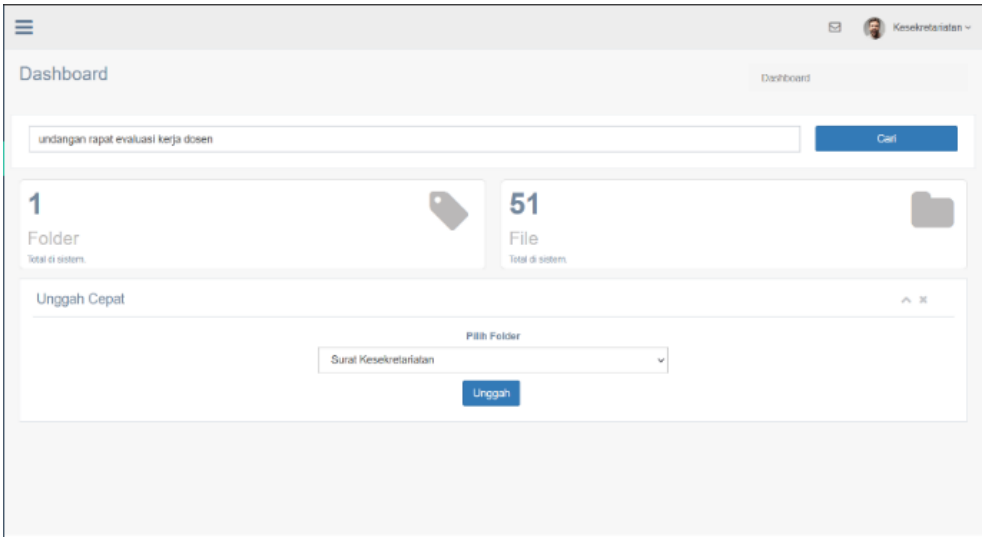

Figure 3 Interface Search Page

Figure 4. Interface Result Search Page

Search page

This page is used by the user to enter a query to find the desired document. Shown in figure 3.

Search Results page

This page displays the documents most similar to the keywords the user has entered on the search page. Shown in figure 4.

Testing

The search result test is done by entering 5 different keywords and the resulting document will determine the threshold value with a cosine similarity score (Francq, 2014). If the document similarity value is less than the threshold value then the document will not appear, conversely if the document similarity value exceeds the document threshold value will appear and as an evaluation a matching will be carried out by the expert (secretary staff) then the recall, precision and accuracy values are calculated using a confusion matrix (Alkaff, 2020). The results of the calculation of the confusion matrix can be seen in table 6.

Table 6 shows that the combination of the weighted tree similarity method with tanimoto cosine has an average recall of 58%, 88% precision, and 83% accuracy, while the combination of weighted tree similarity with cosine sorensen-

Table 6 Confusion Matrix Calculation Results

No	Query	TC			CS		
		recall	precision	accuracy	recall	precision	accuracy
1	pmbk seleksi bantuan beasiswa mahasiswa baru disposisi	50%	100%	83%	60%	100%	87%
2	perkuliahan jadwal libur kuliah dan persiapan daring disposisi	60%	100%	87%	60%	86%	83%
3	kinerja lampiran hasil evaluasi rencana strategi lppm disposisi	30%	75%	73%	60%	86%	83%
4	pmbk pengumuman pendaftaran penerimaan mahasiswa disposisi	70%	78%	83%	70%	78%	83%
5	kinerja sosialisasi kegiatan pembelajaran dosen disposisi	80%	89%	90%	80%	89%	90%
Average		58%	88%	83%	66%	88%	85%

dice has an average recall value of 66%, precision 88%. and accuracy 85%. Combination of weighted tree similarity with cosine sorensen-dice has better recall and accuracy values.

The difference of score accuracy in several query because the search results cosine sorensen-dice found more documents related to the query so that the accuracy value of some queries is higher.

CONCLUSION

The From the results of the analysis and discussion that has been carried out, it can be concluded that the combination of the weighted tree similarity method with the cosine sorensen-dice results in a better search for documents at the University of Muhammadiyah Gresik with a average recall value of 66% and an average accuracy of 85% which is higher than the combination of weigthed tree similarity with tanimoto cosine. And the similarity value on text labels using cosine sorensen-dice is also influenced by the number of terms and documents in the repository. For the development of this research, it can be done by adding a synonym detection method for a word contained in the leaf node.

REFERENCES

Anugrah, I. G., & Sarno, R. (2017). Business Process model similarity analysis using hybrid PLSA and WDAG methods. *Proceedings of 2016 International Conference on Information and Communication Technology and Systems, ICTS 2016*, 231–236. https://doi.org/10.1109/ICTS.2016.7910304

Basmalah Wicaksono, V., Saptono, R., & Widya Sihwi, S. (2016). Analisis Perbandingan Metode Vector Space Model dan Weighted Tree Similarity dengan Cosine Similarity pada kasus Pencarian Informasi Pedoman Pengobatan Dasar di Puskesmas. *Jurnal Teknologi & Informasi ITSmart*, *4*(2), 73. https://doi.org/10.20961/its.v4i2.1768

Alkaff, M., Khatimi, H., & Eriadi, A. (2020). Sistem Rekomendasi Buku Menggunakan Weighted Tree Similarity d an Content Based Filtering. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, *20*(1), 193–202. https://doi.org/10.30812/matrik.v20i1.617

Adi P, P., & Palgunadi, Y. S. (2014). Analisis Kombinasi Algoritma Weighted Tree Similarity dengan Tanimoto Cosine (TC) untuk Pencarin Semantik pada Portal Jurnal. *Prosiding SNST Ke-5 Tahun 2014 Fakultas Teknik Universitas Wahid Hasyim Semarang 1*, 1–6.

Putro, A. N., & Thamrin, H. (2018). Hubungan Fungsi Similaritas dan Persepsi Penutur dalam Menentukan Skor Similaritas Teks Bahasa Indonesia *Skripsi Thesis, Universitas Muhammadiyah Surakarta.*

Anna, & Hendini, A. (2018). Implementasi Vector Space Model Pada Sistem Pencarian Mesin Karaoke. *Evolusi : Jurnal Sains Dan Manajemen*, *6*(1), 1–6. https://doi.org/10.31294/evolusi.v6i1.3535

Suprianto, Sunardi, & Fadlil, A. (2018). Aplikasi Sistem Temu Kembali Angket Mahasiswa

Menggunakan Metode Generalized Vector Space Model. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *6*(1), 33–40. https://doi.org/10.25126/jtiik.201961184

Muttaqin, F. A., & Bachtiar, A. M. (2016). Implementasi Teks Mining pada Aplikasi Pengawasan Penggunaan Internet Anak "DODO KIDS BROWSER" *Jurnal Ilmiah Komputer Dan Informatika*.

Suharso, W., A'yun, Q., & Arifianto, D. (2017). Pengembangan Sistem Deteksi Kesesuaian Dokumen Proposal Program Kreativitas Mahasiswa Dengan Metode Extended Weighted Tree Similarity. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, *2*(2), 84–91. http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO/article/view/1044

Amrizal, V. (2018). Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim). *Jurnal Teknik Informatika*, *11*(2), 149–164. https://doi.org/10.15408/jti.v11i2.8623

Saputra, N., Adji, T. B., & Permanasari, A. E. (2015). Analisis Sentimen Data Presiden Jokowi dengan Preprocessing Normalisasi dan Stemming Menggunakan Metode Naive Bayes dan SVM. *Jurnal Dinamika Informatika*, *5*(November), 12.

Maarif, A. A. (2015). Penerapan Algoritma TF-IDF untuk Pencarian Karya Ilmiah. *Dokumen Karya Ilmiah | Tugas Akhir | Program Studi Teknik Informatika - S1 | Fakultas Ilmu Komputer | Universitas Dian Nuswantoro Semarang*, *5*, 4. mahasiswa.dinus.ac.id/docs/skripsi/jurnal/15309.pdf

Wahyuni, R. T., Prastiyanto, D., & Supraptono, E. (2017). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro*, *9*(1), 18–23. https://journal.unnes.ac.id/nju/index.php/jte/article/view/10955/6659

Lisangan, E. A. (2013). Implementasi n-Gram Technique Dalam Deteksi Plagiarism Pada Tugas Mahasiswa. *TEMATIKA, Journal of Informatics and Information Systems*, *1*(2), 24–30. https://tematika.uajm.ac.id/index.php/tematika/article/view/10

Supriadi, C., Purnomo, H. D., & Sembiring, I. (2020). Sensitivitas Sistem Pencarian Artikel Bahasa Indonesia Menggunakan Metode n-gram Dan Tanimoto Cosine. *Jurnal Transformatika*, *18*(1), 63. https://doi.org/10.26623/transformatika.v18i1.2184

Andika, L. A., Azizah, P. A. N., & Respatiwulan, R. (2019). Analisis Sentimen Masyarakat terhadap Hasil Quick Count Pemilihan Presiden Indonesia 2019 pada Media Sosial Twitter Menggunakan Metode Naive Bayes Classifier. *Indonesian Journal of Applied Statistics*, *2*(1), 34. https://doi.org/10.13057/ijas.v2i1.29998

Francq, Pascal. (2014). Re: Determination of threshold for cosine similarity score?. Retrieved from: https://www.researchgate.net/post/Determination-of-threshold-for-cosine-similarity-score/52ea1b58d5a3f2c3768b45fd/citation/download.