

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/325117028>

Latent Semantic Indexing for Indonesian Text Similarity

Article · March 2018

DOI: 10.14419/ijet.v7i2.3.12619

CITATION

1

READS

108

11 authors, including:



Robbi Rahim

Sekolah Tinggi Ilmu Manajemen Sukma, Medan, Indonesia

175 PUBLICATIONS 606 CITATIONS

[SEE PROFILE](#)



Nuning Kurniasih

Universitas Padjadjaran

78 PUBLICATIONS 93 CITATIONS

[SEE PROFILE](#)



Muhammad Dedi Irawan

Universitas Asahan, Indonesia, Kisaran

5 PUBLICATIONS 1 CITATION

[SEE PROFILE](#)



Deffi Sari

Universitas Bakrie

11 PUBLICATIONS 25 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



A Fast Induction Motor Speed Estimation based on Hybrid Particle Swarm Optimization (HPSO) [View project](#)



Lecturers' Understanding on Indexing Databases of SINTA, DOAJ, Google Scholar, SCOPUS, and Web of Science: A Study of Indonesians [View project](#)

Latent Semantic Indexing for Indonesian Text Similarity

Robbi Rahim^{1*}, Nuning Kurniasih², Muhammad Dedi Irawan³, Yustria Handika Siregar³, Abdurrozzaq Hasibuan⁴, Deffi Ayu Puspito Sari⁵, Tiarna Simaniburuk⁶, Dian Utami Sutiksno⁷, Erland Mouw⁸, Idris Sudin⁹, Achmad Daengs GS¹⁰

¹School of Computer and Communication Engineering, Universiti Malaysia Perlis, Kubang Gajah, Malaysia

²Faculty of Communication Science, Library and Information Science Program, Universitas Padjadjaran, Bandung, Indonesia

³Department of Informatics, Universitas Asahan, Kisaran, Indonesia

⁴Department of Industry Engineering, Universitas Islam Sumatera Utara, Medan, Indonesia

⁵Department of Environmental Engineering, Universitas Bakrie, Jakarta, Indonesia

⁶Department of Information System, STMIK IBBI, Medan, Indonesia

⁷Politeknik Negeri Ambon, Ambon, Indonesia

⁸Universitas Halmahera, Tobelo, Indonesia

⁹Universitas Nuku, Tidore, Indonesia

¹⁰Universitas 45 Surabaya, Surabaya, Indonesia

*Corresponding author E-mail: usurobbi85@zoho.com

Abstract

Document is a written letter that can be used as evidence of information. Plagiarism is a deliberate or unintentional act of obtaining or attempting to obtain credit or value for a scientific work, citing some or all of the scientific work of another party acknowledged as a scientific work without stating the source properly and adequately. Latent Semantic Indexing method serves to find text that has the same text against from a document. The algorithm used is TF/IDF Algorithm that is the result of multiplication of TF value with IDF for a term in document while Vector Space Model (VSM) is method to see the level of closeness or similarity of word by way of weighting term.

Keywords : Similarity, Latent Semantic Indexing, Vector Space Model

1. Introduction

Similarity of sentences may occur either accidentally or intentionally (plagiarism), the examination of the resemblance of the current sentence can be done using tools such as Turnitin, iThenticate or PlagiarismCheckerX or can also search directly by search engines like Google by entering the sentence user want to search, the general concept of examination of sentence similarity is to search and compare process[1]–[4] and this process could be done by using algorithm like Boyer-Moore[5], [6], Knuth-Morris-Pratt[7], Breadth-First Search[8], Depth-First Search[9] or any other algorithm.

The publication of scientific papers, academic articles, these are fragile documents that will result in similarities to the detriment of many parties[10]–[12], to examine the possibilities that can occur it in analysis by using Latent Semantic Indexing (LSI) method[13]–[15]. The LSI method is a statistical method that extracts the semantic structure of a word or phrase from a document, if the number of common words in the document is very large then the sentence is semantic. The LSI method will summarize the existing sentence and then calculate to show the calculation of the similarity of words that may be contained in the document[13], [14], as well as the application of the Term Frequency/Inverse Document Frequency algorithm to calculate the frequency of occurrence of term from a document[16]–[20].

The application [21]–[23] of the LSI method of examining the resemblance of sentences may help to avoid accidental or deliberate plagiarism and may be used for some other examinations such as the repetition of words that may be contained in a sentence contained in the document.

2. Methodology

Process examination with Latent Semantic Indexing algorithm[10], [13], [14] can be perform in several steps as follows:

- Calculating the value of the term frequency (tf).
- Calculate the value of the document frequency (df).
- Calculating inverse document frequency (idf).
- Calculate the weight of the document (W).
- Calculate the multiplication of the value of the keyword weight (query) of the document (WD) with the weight value of the i document (Wdi), sum the result of the multiplication of the weight value.
- Calculate the vector length of each document and keyword (query).
- Calculate similarity.

The final result is then calculated the level of similarity with the keyword (query) using vector space model calculation.

3. Results and Discussion

Implementation phase Latent Semantic Indexing TF/IDF to be more clearly created examples of keywords (Q = query) and documents (D = 4), as below:

Keyword (Q) = Analisis pada dokumen otomatis menggunakan metode LSI

Document 1 (D1)= Sistem analisis dokumen teks

Document 2 (D2)= Penilaian sistem dokumen teks terhadap dokumen otomatis

Document 3 (D3)= dalam sistem dokumen teks pada algoritma TF/IDF dan LSI

Document 4 (D4)= Analisis sistem pada dokumen otomatis untuk proses clustering dokumen menggunakan LSI dan TF/IDF

Through tokenizing process then entering the filtering process (stopword and stoplist), the word "pada" in Q, the word "terhadap" on D2, the word "dalam", "pada" and the "/" in D3, the word "pada" "untuk" and "dan" on D4 are deleted. Furthermore, a collection of words that have been selected are processed by the weighting of documents through the following calculations.

- a. Calculating the value of term frequency (tf). Table 1 shows TF Value.

Table 1: TF Value

No.	Term	TF				
		Q	D1	D2	D3	D4
1	Analisis	1	1	0	0	1
2	Dokumen	1	1	2	1	2
3	Otomatis	1	0	1	0	1
4	Menggunakan	1	0	0	0	1
5	LSI	1	0	0	1	1
6	Sistem	0	1	1	1	1
7	Teks	0	1	1	1	0
8	Penilaian	0	0	1	0	0
9	Algoritma	0	0	0	1	0
10	Tf	0	0	0	1	0
11	Idf	0	0	0	1	0
12	Proses	0	0	0	0	1
13	Clustering	0	0	0	0	1

After the tokenizing process is done, it can be the result of the term frequency (TF) value of each word in the sentence and form a matrix on each document. Next calculate the value of the document frequency (DF).

- b. Calculates the value of the document frequency (DF). Document frequency (DF) is the number of documents in which a word (term) appears. Table 2 shows DF value.

Table 2: DF Value

No.	Term	DF
1	Analisis	2
2	Dokumen	4
3	Otomatis	2
4	Menggunakan	1
5	LSI	2
6	Sistem	4
7	Teks	3
8	Penilaian	1
9	Algoritma	1
10	Tf	1
11	Idf	1
12	Proses	1
13	Clustering	1

After the calculation results TF and DF obtained, the next step calculation inverse document frequency (IDF) of each word (term) to calculate the weight of the word (term).

- c. Calculate inverse document frequency (IDF)

Formula used: $IDF = \log \frac{D}{DF}$

Information:

D = number of documents

Through IDF calculation obtained IDF calculation results.

Table 3: IDF Value

Term	Tf					df	D/df	IDF
	Q	D1	D2	D3	D4			
Analisis	1	1	0	0	1	2	4/2	0,301
Dokumen	1	1	2	1	2	4	4/4	0
Otomatis	1	0	1	0	1	2	4/2	0,301
Menggunakan	1	0	0	0	1	1	4/1	0,602
LSI	1	0	0	1	1	2	4/2	0,301
Sistem	0	1	1	1	1	4	4/4	0
Teks	0	1	1	1	0	3	4/3	0,124
Penilaian	0	0	1	0	0	1	4/1	0,602
Algoritma	0	0	0	1	0	1	4/1	0,602
Tf	0	0	0	1	0	1	4/1	0,602
Idf	0	0	0	1	0	1	4/1	0,602
Proses	0	0	0	0	1	1	4/1	0,602
Clustering	0	0	0	0	1	1	4/1	0,602

Table 3 is an IDF computation process with different values for every word token in process, where the value 0 is a word token with no resemblance. Furthermore, after the tf and idf values have been obtained, it is then included in the calculation of tf-idf weighting to calculate the weight of the relationship of a term (term) in the document.

- d. Calculates the weight of the document (W)

Formula used: $W_{d,t} = tf_{d,t} * IDF$

Information:

d=d document

t= the t word of the keyword

W= the weight of the d document against the t-word

Table 4: W value (document weight)

No.	Term	tf					IDF=log (D/df)	W _{dt} =tf _{dt} *IDF				
		Q	D1	D2	D3	D4		Q	D1	D2	D3	D4
1	Analisis	1	1	0	0	1	0,301	0,301	0,301	0	0	0,301
2	Dokumen	1	1	2	1	2	0	0	0	0	0	0
3	Otomatis	1	0	1	0	1	0,301	0,301	0	0,301	0	0,301
4	Menggunakan	1	0	0	0	1	0,602	0,602	0	0	0	0,602
5	LSI	1	0	0	1	1	0,301	0,301	0	0	0,301	0,301
6	Sistem	0	1	1	1	1	0	0	0	0	0	0
7	Teks	0	1	1	1	0	0,124	0	0,124	0,124	0,124	0
8	Penilaian	0	0	1	0	0	0,602	0	0	0,602	0	0
9	Algoritma	0	0	0	1	0	0,602	0	0	0	0,602	0
10	Tf	0	0	0	1	0	0,602	0	0	0	0,602	0
11	Idf	0	0	0	1	0	0,602	0	0	0	0,602	0
12	Proses	0	0	0	0	1	0,602	0	0	0	0	0,602
13	Clustering	0	0	0	0	1	0,602	0	0	0	0	0,602

Table 4 is the weighting for each document based on a word token that has been processed by obtaining the IDF value of each word, the weighting is grouped in Q, D1, D2, D3 and D4.

Furthermore, after calculating the multiplication of $WD * W_{d_i}$, the sum of the results of the multiplication of the value of the weight is shown in Table 6.

- e. Calculates the multiplication of the value of the document weight (WD) with the weight value of the i th document (W_{d_i}) as shown in Table 5.

Table 5: Multiplication $WD * W_{d_i}$

WD*W _{d_i}			
D1	D2	D3	D4
0,091	0	0	0,091
0	0	0	0
0	0,091	0	0,091
0	0	0	0,362
0	0	0,091	0,091
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Table 6: $WD * W_{d_i}$ sum

Document	Function	Result
	$SUM(WD * W_{d_i}) = \sum_{j=1}^n WD_j W_{d_{i,j}}$	
D1	$SUM(WD * W_{d_1}) = \sum_{j=1}^n WD_j W_{d_{1,j}}$	$0,091 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 0,091$
D2	$SUM(WD * W_{d_2}) = \sum_{j=1}^n WD_j W_{d_{2,j}}$	$0 + 0 + 0,091 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 0,091$
D3	$SUM(WD * W_{d_3}) = \sum_{j=1}^n WD_j W_{d_{3,j}}$	$0 + 0 + 0 + 0 + 0,091 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 0,091$
D4	$SUM(WD * W_{d_4}) = \sum_{j=1}^n WD_j W_{d_{4,j}}$	$0,091 + 0 + 0,091 + 0,362 + 0,091 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 0,634$

After the multiplication of $WD * W_{d_i}$ and sum of $WD * W_{d_i}$ is obtained, then calculate the length of each document vector and keyword (query).

- f. Calculates the vector length of each document and keyword (query) as shown in Table 7 and Table 8.

Table 7: Vector calculation from query

Query	Function	Result
	$ Q = \sqrt{\sum_i W_{q,j}^2}$	
Q	$sqrt(Q) = sqrt(\sum_{j=1}^n Q_j^2)$	$= \sqrt{(0,301)^2 + (0)^2 + (0,301)^2 + (0,602)^2 + (0,301)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2}$ $= \sqrt{0,091 + 0 + 0,091 + 0,362 + 0,091 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0}$ $= \sqrt{0,635}$ $= 0,796$

Table 8: Vector calculation from document

Document	Function		Result
	$ D_i = \sqrt{\sum_i W_{i,j}^2}$		
D1	$sqrt(Q) = sqrt\left(\sum_{j=1}^n D_{1,j}^2\right)$		$= \sqrt{(0,301)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0,124)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2}$ $= \sqrt{0,091 + 0 + 0 + 0 + 0 + 0 + 0 + 0,124 + 0 + 0 + 0 + 0 + 0}$ $= \sqrt{0,106}$ $= 0,326$
D2	$sqrt(D_2) = sqrt\left(\sum_{j=1}^n D_{2,j}^2\right)$		$= \sqrt{(0)^2 + (0)^2 + (0,301)^2 + (0)^2 + (0)^2 + (0)^2 + (0,124)^2 + (0,602)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2}$ $= \sqrt{0 + 0 + 0,091 + 0 + 0 + 0 + 0 + 0,015 + 0,362 + 0 + 0 + 0 + 0 + 0}$ $= \sqrt{0,468}$ $= 0,684$
D3	$sqrt(D_3) = sqrt\left(\sum_{j=1}^n D_{3,j}^2\right)$		$= \sqrt{(0)^2 + (0)^2 + (0)^2 + (0)^2 + (0,301)^2 + (0)^2 + (0,124)^2 + (0)^2 + (0,602)^2 + (0,602)^2 + (0)^2 + (0)^2 + (0)^2}$ $= \sqrt{0 + 0 + 0 + 0 + 0,091 + 0 + 0,015 + 0 + 0,362 + 0,362 + 0,362 + 0 + 0}$ $= \sqrt{1,192}$ $= 1,092$
D4	$sqrt(D_4) = sqrt\left(\sum_{j=1}^n D_{4,j}^2\right)$		$= \sqrt{(0,301)^2 + (0)^2 + (0,301)^2 + (0,602)^2 + (0,301)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0,602)^2 + (0,602)^2}$ $= \sqrt{0,091 + 0 + 0,091 + 0,362 + 0,091 + 0 + 0 + 0 + 0 + 0 + 0 + 0,362 + 0,362}$ $= \sqrt{1,359}$ $= 1,166$

After the calculation of vector query and document, then get the value of vector length as shown in Table 9.

Table 9: Vector Length Values

Vector Length Values				
Q	D1	D2	D3	D4
0,091	0,091	0	0	0,091
0	0	0	0	0
0,091	0	0,091	0	0,091
0,362	0	0	0	0,362
0,091	0	0	0,091	0,091
0	0	0	0	0
0	0,015	0,015	0,015	0
0	0	0,362	0	0
0	0	0	0,362	0
0	0	0	0,362	0
0	0	0	0,362	0
0	0	0	0	0,362
0	0	0	0	0,362

The next step is to calculate the Similarity between the keyword vectors (query) with each document.

g. Counting Similarities is shown in Table 10.

Table 10: Counting Similarities

Document	Function	Result	
	$Cosine\theta_{D_i} = \frac{Q, D_i}{ Q * D_i }$		
D1	$Cosine\theta_{D_1} = \frac{Q, D_1}{ Q * D_1 }$	$\frac{0,091}{0,796 * 0,326} = \frac{0,091}{0,259}$	$= 0,349$
D2	$Cosine\theta_{D_2} = \frac{Q, D_2}{ Q * D_2 }$	$\frac{0,091}{0,796 * 0,684} = \frac{0,091}{0,545}$	$= 0,166$
D3	$Cosine\theta_{D_3} = \frac{Q, D_3}{ Q * D_3 }$	$\frac{0,091}{0,796 * 1,092} = \frac{0,091}{0,87}$	$= 0,104$
D4	$Cosine\theta_{D_4} = \frac{Q, D_4}{ Q * D_4 }$	$\frac{0,634}{0,796 * 1,166} = \frac{0,634}{0,928}$	$= 0,683$

Cosine calculation results note that Document 4 (D4) has the highest similarity level followed by D1, D2 and D3.

4. Conclusion

Examination of similarity of words or sentences by using Latent Semantic Indexing can be done well, the process of examination similarity done enough detail so that the margin error obtained is also small, the development by combining with other algorithms is possible so that the results obtained will also be better.

References

- [1] R. Rahim, S. Nurarif, M. Ramadhan, S. Aisyah, and W. Purba, "Comparison Searching Process of Linear, Binary and Interpolation Algorithm," *J. Phys. Conf. Ser.*, vol. 930, no. 1, p. 12007, Dec. 2017.
- [2] D. Abdullah, R. Rahim, D. Apdilah, S. Efendi, T. Tulus, and S. Suwilo, "Prime Numbers Comparison using Sieve of Eratosthenes and Sieve of Sundaram Algorithm," in *Journal of Physics: Conference Series*, 2018, vol. 978, no. 1, p. 12123.
- [3] J. Sastry, M. Sri Harsha Vamsi, R. Srinivas, and G. Yeshwanth, "Optimizing performance of search engines based on user behavior," vol. 7, no. 2.7 Special Issue 7, pp. 359–362, 2018.
- [4] R. Rahim *et al.*, "Searching Process with Raita Algorithm and its Application," *J. Phys. Conf. Ser.*, vol. 1007, no. 1, p. 12004, Apr. 2018.
- [5] R. Rahim, A. S. Ahmar, A. P. Ardyanti, and D. Nofriansyah, "Visual Approach of Searching Process using Boyer-Moore Algorithm," *J. Phys. Conf. Ser.*, vol. 930, no. 1, p. 12001, Dec. 2017.
- [6] D. N. K. Manjit Jaiswal, "An Enhanced Boyer-Moore Algorithm for Text String Matching Problem," *GSTF J. Comput.*, vol. 2, 2012.
- [7] R. Rahim, I. Zulkarnain, and H. Jaya, "A review: search visualization with Knuth Morris Pratt algorithm," in *IOP Conference Series: Materials Science and Engineering*, 2017, vol. 237, no. 1, p. 12026.
- [8] R. Ratnadewi, E. M. Sartika, R. Rahim, B. Anwar, M. Syahril, and H. Winata, "Crossing Rivers Problem Solution with Breadth-First Search Approach," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 288, no. 1.
- [9] R. Rahim *et al.*, "Block Architecture Problem with Depth First Search Solution and Its Application," *J. Phys. Conf. Ser.*, vol. 954, no. 1, p. 12006, 2018.
- [10] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Trans. Knowl. Discov. Data*, vol. 2, no. 2, pp. 1–25, 2008.
- [11] D. Stumpfe and J. Bajorath, "Similarity searching," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 1, no. 2, pp. 260–282, 2011.
- [12] R. Rahim *et al.*, "INA-Rxiv: The Missing Puzzle in Indonesia's Scientific Publishing Workflow," *J. Phys. Conf. Ser.*, vol. 1007, no. 1, p. 12032, Apr. 2018.
- [13] S. Deerwester, S. T. Dumais, and R. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.
- [14] A. Kontostathis, "Essential dimensions of latent semantic indexing (LSI)," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2007.
- [15] R. Blessy Jenila and S. Bharathi, "Recommendation on semantic web pages based on conceptual prediction model," vol. 7, no. 1.7 Special Issue 7, pp. 199–200, 2018.
- [16] D. Metzler, "Generalized Inverse Document Frequency," in *Proceeding of the 17th ACM conference on Information and knowledge management*, 2008, pp. 399–408.
- [17] M. Nurjannah, H. Hamdani, and I. F. Astuti, "Penerapan Algoritma Term Frequency-Inverse Document Frequency (TF-IDF) untuk Text Mining," *J. Inform. Mulawarman*, vol. 8, no. 3, pp. 110–113, 2016.
- [18] R. Rahim, D. Hartama, H. Nurdianto, A. S. Ahmar, D. Abdullah, and D. Napitupulu, "Keylogger Application to Monitoring Users Activity with Exact String Matching Algorithm," *J. Phys. Conf. Ser.*, vol. 954, no. 1, p. 12008, 2018.
- [19] R. Rahim, Nurjamiyah, and A. R. Dewi, "Data Collision Prevention with Overflow Hashing Technique in Closed Hash Searching Process," *J. Phys. Conf. Ser.*, vol. 930, no. 1, p. 12012, Dec. 2017.
- [20] R. Radhika Jahnvi.S, K. Kumar.K, and S. Hareesh.T, "A Semantic web based filtering techniques through web service recommendation," *Int. J. Eng. Technol.*, vol. 7, no. 2–7, p. 41, Mar. 2018.
- [21] D. Napitupulu *et al.*, "Analysis of Student Satisfaction Toward Quality of Service Facility," *J. Phys. Conf. Ser.*, vol. 954, no. 1, p. 12019, Jan. 2018.
- [22] A. S. Ahmar *et al.*, "Lecturers' understanding on indexing databases of SINTA, DOAJ, Google Scholar, SCOPUS, and Web of Science: A study of Indonesians," *J. Phys. Conf. Ser.*, vol. 954, 2018.
- [23] A. Phani Sheetal and K. Ravindranath, "Software metric evaluation on cloud based applications," *Int. J. Eng. Technol.*, vol. 7, no. 1.5 Special Issue 5, pp. 13–18, 2018.