TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 1, February 2020, pp. 217~227
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i1.14874

Latent semantic analysis and cosine similarity for hadith search engine

Wahyudin Darmalaksana[1], Cepy Slamet[2], Wildan Budiawan Zulfikar[3], Imam Fahmi Fadillah[4], Dian Sa'adillah Maylawati[5], Hapid Ali[6]
[1]Department of llmu Hadis, UIN Sunan Gunung Djati Bandung, Indonesia
[2,3,4,5]Department of Informatics, UIN Sunan Gunung Djati Bandung, Indonesia
[2]Department of Infomation and Communication Technology, Asia e University, Malaysia
[5]Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Malaysia
[6]Faculty of Tarbiyah and Education, UIN Sunan Gunung Djati Bandung, Indonesia

Article Info

Article history:

Received Aug 14, 2019
Revised Dec 5, 2019
Accepted Dec 22, 2019

Keywords:

Cosine similarity
Hadith
Latent semantic analysis
Search engine

ABSTRACT

Search engine technology was used to find information as needed easily, quickly and efficiently, including in searching the information about the hadith which was a second guideline of life for muslim besides the Holy Qur'an. This study was aim to build a specialized search engine to find information about a complete and eleven hadith in Indonesian language. In this research, search engines worked by using latent semantic analysis (LSA) and cosine similarity based on the keywords entered. The LSA and cosine similarity methods were used in forming structured representations of text data as well as calculating the similarity of the keyword text entered with hadith text data, so the hadith information was issued in accordance with what was searched. Based on the results of the test conducted 50 times, it indicated that the LSA and cosine similarity had a success rate in finding high hadith information with an average recall value was 87.83%, although from all information obtained level of precision hadith was found semantically not many, it was indicated by the average precision value was 36.25%.

Corresponding Author:

Dian Sa'adillah Maylawati,
Department of Informatics,
UIN Sunan Gunung Djati Bandung,
Indonesia.
Email: diansm@uinsgd.ac.id

1. INTRODUCTION

Search engine becames one of functions or the most important tool on information system specially on-line system [1]. Search engine technology gives it easy for system user to get the information quickly [2]. Google is one of capable search engines but it still has limitations in analyzing the content and meaning of search results [3]. Along with advanced date regulation on the internet, search engines require speed and accuracy in releasing results in line with expectations today. The search function becomes important thing in getting information easily and quickly. However, not all search engines are devoted to find certain information precisely and accurately. In this study, a search engine that was built specifically to get information about the hadith in accordance with user needs. Where, the hadith is the second important source of law for Muslims after the Holy Qur'an [4, 5]. Of course, the generated hadith information must hand in hand with needed requirements. Therefore, search engines that are built need to consider the semantics wheather from the inputted keywords or the hadith data which is saved in the system.

Hadith collection in the form of text requires certain processes so that the meaning of the text is maintained [6]. Starting from preparing unstructured text data into structured data [7, 8]. Structured representation of text can be used in the next processes both in information retrieval (IR) and text mining [9]. In the study of obtained information search engine, it uses the information retrieval (IR) technique by combining the latent semantic analysis algorithm and cosine similarity. In contrast to text mining where the results obtained from the system are not clear yet, IR produces information that has actually been known its form, because it is the same as the collection of data held [10–12]. Information retrieval (IR) is used to connect relationships between large text data collections according to keywords. The parts of IR include:

- Text operations (operations of text) which include the selection of words in keywords or documents (term selection) in the transformation of documents or keywords become term indexes (index of words).
- Query formulation (formulation of keywords) that gives a standard to the word indexes of keyword.
- Ranking (ranking), look for documents that are relevant to keywords and arrange the documents according to their compatibility with keywords.
- Indexing (indexing), build a data base of indexes from document collections. Firstly, it is carried before searching documents.

IR system accepts keywords from users, then ranks documents on collections based on their compatibility with keywords. The result of rank which is given to users is documents based on the system are relevant to keywords. But the relevance of documents to a keyword is a subjective judgment and it is influenced by many factors such as topics, timing, sources of information and the objective of users.

Latent semantic analysis algorithm is widely used in processing text data by semantics approaches so the meaning of the text is maintained. Latent semantic analysis can be used not only for text summarization well [13–15], checking plagiarism [15], and automatically evaluating essays [16], of course it can also be used for searching. Latent semantic analysis compares the entered text with owned text data collection based on vector representations [17–19], with regard to semantics approaches to preserve the meaning of texts. In addition to latent semantic analysis, this hadith search engine research also uses cosine similarity to see the similarity of text data generated by search engines so that it can bring up text data sequences based on popularity as top order. Cosine similarity is one of the most popular similarity calculation methods to be applied to text documents [20]. The main advantage of the cosine similarity method is that it can't be affect by the length and short of a document. Because the term value of each document is the important thing. Based on the explanation of the problem formulation above, how latent semantic analysis and cosine similarity can be implemented in finding the hadith text based on keywords entered correctly on the hadith search engine? Are latent semantic analysis and cosine similarity in the search engine can find hadith text data that are searched based on keywords that are entered correctly and relevant.

2. RESEARCH METHOD

Figure 1 describes activity flow of this research. Generally, this reseach used IR technique that implement latent semantic analysis and cosine similarity algorithm for producing information of hadiths based on input keywords. The activity begin from inputing the keywords (can be in the form of words, phrase, or sentence), the input keyword will be processed in text pre-processing phase to clean text data. Then, LSA agorithm will be conducted to create term document matrix and get the vector value of each document. Last, the similarity of input keywords and hadith data collection will be counted using cosine similarity.

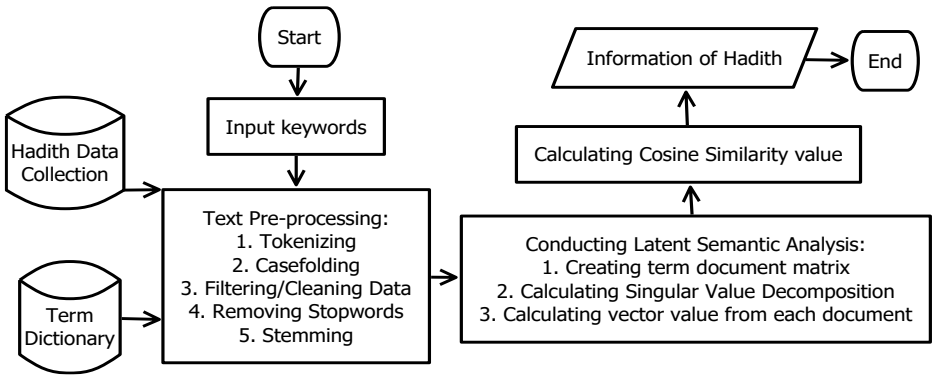

Figure 1. Research Activities

2.1. Latent semantic analysis (LSA)

Latent semantic analysis is an algebraic method that extracts hidden semantic structures from words and sentences [21]. Latent semantic analysis algorithm is one of the development algorithms in the field of information retrieval that is able to collect a large number of documents in a data base and connect relationships between documents by matching the given input. The main function of this latent semantic analysis is to calculate the similarity of a text data by comparing vector representations from other text data [15]. The results of latent semantic analysis represent text data contextually and semantic that gives text meanings [21, 22]. The evaluation by using the latent semantic analysis method focuses on words in writing without considering to the order of words and grammar in writteng texts so that a sentence is assessed based on the key words include in the sentence [23]. Basically, latent semantic analysis extracts information from patterns or collections of words that often appear simultaneously in different sentences. If the sentence contains a collection of words that often appear in large numbers, the sentence has semantic or safe meaning [21]. Generally, the steps of latent semantic analysis that are used for text data, among others [24]: text pre-processing, creating term of document matrix, calculating singular value decomposition (SVD) and calculating vector value for each document

2.1.1. Text pre-processing

The text pre-processing stage is the stage to prepare text data which is unstructured data becomes a structured data representation [7, 25, 26]. The process starts from tokenization, deletes regular expressions, deletes non letter characters, deletes stop words, and stemming. In fact, if needed, it is carried out a special process to handle natural languages contained in text data, such as; abbreviations, slang, regional languages, and other natural languages. The discussion regarding text pre-processing will be explained further in section 3.2.

2.1.2. Creating term of document matrix

After carried out the pre-processing stage in the text data, then the term of document matrix is constructed by placing the word result of the stemming (term) process into the row. This matrix is called the term of document matrix. Each row represents a unique word, while each column represents the obtained word source. The source of the word can be sentences, paragraphs, or all parts of the text. The examples of the term of document matrix can be seen in Table 1 (that presented with Indonesian language). On the Table 1, the first row represents the word has passed the pre process until the stemming process is called stemmed term (the word as term 1, term 2, etc.), and the column represents the context, namely the text. The value is located in each cell on the table shows how the number of times in a term appears in a document. For instance, the term 1 appears 1 time at the firts document, and appears 2 times at the second document, but the term 1 does not appear at third document, and so on.

Table 1. Matrix example for term of document

Word	Doc 1	Doc 2	Doc 3
jangan (do not)	1	1	0
kalian (you)	1	1	0
dusta (lie)	1	1	1
atas (on behalf)	1	1	1
nama (name)	1	1	1
niscaya (surely)	1	0	0
masuk (enter)	1	1	0
neraka (the hell)	1	1	1
sungguh (actually)	0	1	0
sengaja (expressly)	0	0	1
tempat (place)	0	0	1
duduk (seat)	0	0	1
hendak (should)	0	0	1

2.1.3. Calculating singular value decompsition and vector value for each document

Singular value decomposition SVD is a linear algebra theorem which can split term of document matrix into three new matrices, those are: orthogonal matrix or left singular vector matrix (U), diagonal matrix or singular value matrix (S), and transpose of orthogonal matrix or right singular matrix (V) [27–29], formulated by (1) that illustrated in Figure 2.

$$A = US\,V^T \quad (1)$$

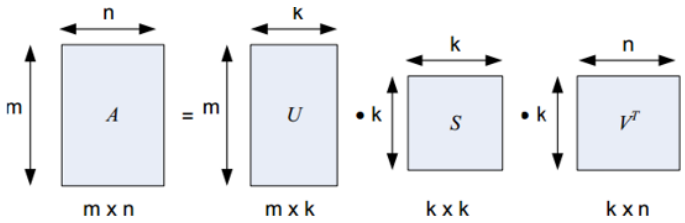

Figure 2. SVD Illustration of (1) [30]

The formula (1) is obtained from the U matrix which is a matrix of m x k size and a matrix V of n x k size, as illustrated in Figure 1, U and V which have orthogonal columns so that it ca be valid:

$$UT\ U = VT\ V = 1 \quad (2)$$

and S is a diagonal matrix of k x k size. The contents on the main diagonal of the S matrix are singular of the A matrix. The results of the SVD can be better understood if A matrix is written with a different interpretation. If $u_{1,}u_{2,\ldots,}u_k$ are column vectors of the U matrix, $\sigma_{1,}\sigma_{2,\ldots}\sigma_k$ are entries in the main diagonal of the S matrix, and $v_{1,}v_{2,\ldots}v_k$ are column vectors of V matrix, A matrix can be written as shown in (3).

$$A = \sum_{i=1}^{k} \sigma_i\ u_i\ v_i^T \quad (3)$$

where the value of σ1 is for 1, for i = 1, 2, ..., k, on (3) it is sorted from the largest to the smallest. If some big values σ_1 are taken and a small (near zero) σ_ (1) value is discarded, we get an approximation from good A value. So, by using SVD, a matrix can be written as a sum of the components ($v_1\ v_i^T$ for i = 1, 2, …, k), and its weight is the singular value (v_1 , for i = 1, 2, … k, are taken from the formula of (4) [30].

$$A = [u_{1,}u_{2,\ldots,}u_k] \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_k^T \end{bmatrix} \quad (4)$$

SVD can identify and arange dimensions that indicate which data variations often appear. SVD takes the term of document matrix which consists of words and documents as in Table 1 which has been broken down into linear independent components. The result of the SVD process is a vector that will be used to be calculated its similarity by an approach.

2.1.4. Calculating cosine similarity

Cosine similarity is used to calculate the cosine value between documents vector in a collection and the needed input vector [31, 32]. The smaller the produced, the higher the level of similarity of the essay occure. The formula of cosine similarity is as shown in (5):

$$\text{Cos } \alpha = \frac{\text{A.B}}{|\text{A}|.|\text{B}|} = \frac{\sum_{i=1}^{n} \text{A}i \text{ x B}i}{\sqrt{\sum_{i=1}^{n} (\text{A}i)^2 \text{x } \sum_{i=1}^{n} (\text{B}i)^2}} \quad (5)$$

with the statement, it showed that A is a document vector, B is an input vector, A. B is the dot product of vector A with vector B, |A| is the length of vector A, |B| is the length of vector B, |A|. |B| is a cross product between |A| and |B| and α is the angel which is formed between vector A and vector B.

3. RESULTS AND ANALYSIS

In this section, it is explained the results of research and at the same time is given the comprehensive discussion about how LSA and CS are implemented in searching information of hadiths and present the evaluation result of experiment that conducted.

3.1. Pre-processing for text data

Text data is unstructured data that needs special treatment before caried out mining process or searching for information contained in the text [30]. The pre processing stage for text is the stage of preparing text data into a structured data representation. Generally, two types of structured data representations for text

are bag of words and multiple of words [33, 34]. Latent semantic analysis is one algorithm that produces structured text representations in the form of multiple of words. Where, the text is not only represented by 1 word but also can be more than 1 word or also known as n-gram. Even the latent semantic analysis word collections considers to the semantics between one word and another.

Pre-processing of text data starts from uniformity of the size of letters to lowercase, deleting characters other than letters and regular expressions, if it is necessary to change abbreviations to be their original form, delete unimportant words or stop word removal, then it is the process to change the initial words into words essentially or stemming. In this study, the stemming process uses the Nazief & Adriani algorithm because the hadith text documents are arranged in Indonesian. The Nazief & Adriani algorithm is the most commonly used stemming algorithm for Indonesian because it is in accordance with the syntax of Indonesian [35–39]. The results of the stemming used as data are entered for the latent semantic analysis and formed the term of document matrix from the text data.

3.2. Implementation of latent semantic analyais and cosine similarity on the hadith search engines

Latent semantic analyais is applied after the pre processof text is complete. Then the pre process results will be formed to be term of document matrix. The term of document matrix will be computed by SVD to produce a matrix of U, S, and V. The final stage is the application of cosine similarity to see the similarity of the information generated as well as arange it based on the level of similarity. The flow of the latent semantic analysis and cosine similarity that impemented in this study can be seen at the Figure 1. For instance, there are 3 pieces of the following hadith documents (present in Indonesian language):

Document 1:
Janganlah kalian berdusta atas namaku, karena siapa yang berdusta atas namaku niscaya dia masuk neraka.
(Do not lie on behalf of my name, because if anyone who lies on behalf of my name, he/she will go to the hell surely.)
Document 2:
Janganlah kalian berdusta terhadapku (atas namaku), karena barangsiapa berdusta terhadapku dia akan masuk neraka.
(Do not lie to me (on my behalf), because whoever lies on me he will go to the hell.)
Document 3:
Barangsiapa yang sengaja melakukan kedustaan atas namaku, maka hendaklah dia menempati tempat duduknya dari neraka.
(Whoever deliberately lies on behalf of my name, he should occupy his seat from the hell.)

Input Keywords in Hadith Search Engine:
Jangan Dusta Masuk Neraka
(Do not lie to go to the hell)

Text data from these three documents and go to the search engine. It will be caried out pre-proccess to produce text data as follows:

Document 1: *jangan kalian dusta atas nama dusta*
Document 2: *jangan kalian dusta atas nama dusta masuk neraka*
Document 3: *sengaja dusta atas nama hendak tempat duduk neraka*
Input keywords in hadith search engine: *jangan dusta masuk neraka*

Then, the already three prepared text data is processed to form matrixes of the term of document likes on Table 1 and it is gained A matrixes as follows:

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

The main step that needs to be completed is to decompose A matrix to be 3 other matrices using SVD, starting from finding the ATA value to calculate with cosine similarity. The process of applying Laten Semantics Analysis and Cosine Similarity for the term of document matrix is in the following Table 1. Search the value of ATA:

$$\text{ATA} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} x \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 8 & 7 & 4 \\ 6 & 7 & 4 \\ 4 & 4 & 8 \end{pmatrix}$$

search determinant of ATA result, so |ATA-λI|=0 :

$$A^T A - \lambda I = \begin{pmatrix} 8 & 7 & 4 \\ 6 & 7 & 4 \\ 4 & 4 & 8 \end{pmatrix} - \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix} = \begin{pmatrix} 8-\lambda & 7 & 4 \\ 6 & 7-\lambda & 4 \\ 4 & 4 & 8-\lambda \end{pmatrix}$$

$$|\text{ATA} - \lambda I| = (8-\lambda)\det\begin{pmatrix} 7-\lambda & 4 \\ 4 & 8-\lambda \end{pmatrix} - (7)\, det\begin{pmatrix} 6 & 4 \\ 4 & 8-\lambda \end{pmatrix} - (4)\, det\begin{pmatrix} 6 & 7-\lambda \\ 4 & 4 \end{pmatrix}$$

$$|\text{ATA} - \lambda I| = [(7)(8-\lambda) - (4)(4)] - (7)[(6)(8-\lambda) - (4)(4)] + (4)\,[(6)(4)\,[-(7-\lambda)(4)]$$

$$|A^T A - \lambda I| = \lambda^3 + 23\lambda^2 - 102\lambda + 80 = 0$$

search eigen value and eigen vactor:

Eigen Value:	Eigen Vactor:
λ1= 17.40312	V1 = 1.24704, 1.10373, 1
λ2= 4.59687	V2= -0.54366, -0.30712, 1
λ3= 1	V3= -1, 1, 0

search singular matrix based on the value of eigen value which has been gained:

$$S1 = \sqrt{17.40312} = 4.1717$$

$$S2 = \sqrt{4.59687} = 2.14403$$

$$S3 = \sqrt{1} = 1$$

$$S = \begin{pmatrix} S1 & 0 & 0 \\ 0 & S2 & 0 \\ 0 & 0 & S3 \end{pmatrix} = \begin{pmatrix} 4.1717 & 0 & 0 \\ 0 & 2.14403 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$S\text{-}1 = \begin{pmatrix} 0.23971 & 0 & 0 \\ 0 & 0.46641 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

search V matrix value by using value normalization of eigen vactor which has been gained:

$$|V1| = \sqrt{1.24704^2 + 1.10373^2 + 1^2} = 1.94251$$

$$|V2| = \sqrt{-0.54366^2 + -0.30712^2 + 1^2} = .17894$$

$$|V3| = \sqrt{-1^2 + 1^2 + 0^2} = 1.41421$$

$$V1 = \frac{1.24704}{1.94251}, \frac{1.10373}{1.94251}, \frac{1}{1.94251} = 0.64197, 0.56819, 0.51479$$

$$V2 = \frac{-0.54366}{1.17894}, \frac{-0.30712}{1.17894}, \frac{1}{1.17894} = -0.46114, -0.26051, 0.84822$$

$$V3 = \frac{-1}{1.41421}, \frac{1}{1.41421}, \frac{0}{1.41421} = -0.70711, 0.70711, 0$$

formulate V matrics with gained value fromthe result of normalization calculation of eigen vactor:

$$V = \begin{pmatrix} 0.64197 & 0.56819 & 0.51479 \\ -0.46114 & -0.26051 & 0.84822 \\ -0.70711 & 0.70711 & 0 \end{pmatrix}$$

$$VT = \begin{pmatrix} 0.64197 & -0.46114 & -0.70711 \\ 0.56819 & -0.26051 & 0.70711 \\ 0.51479 & 0.84822 & 0 \end{pmatrix}$$

search U matrix value with the formula of U= AVS-1:

$$U = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} x \begin{pmatrix} 0.64197 & 0.56819 & 0.51479 \\ -0.46114 & -0.26051 & 0.84822 \\ -0.70711 & 0.70711 & 0 \end{pmatrix} x \begin{pmatrix} 0.23971 & 0 & 0 \\ 0 & 0.46641 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} 0.04335 & 0.14351 & 1.36301 \\ 0.04335 & 0.14351 & 1.36301 \\ -0.12615 & 0.47331 & 1.36301 \\ -0.12615 & 0.47331 & 1.36301 \\ -0.12615 & 0.47331 & 1.36301 \\ 0.15389 & 0.26501 & 0.51479 \\ 0.04335 & 0.14351 & 1.36301 \\ -0.12615 & 0.47331 & 1.36301 \\ -0.11054 & -0.12150 & 0.84822 \\ -0.16590 & 0.32980 & 0 \\ -0.16590 & 0.32980 & 0 \\ -0.16590 & 0.32980 & 0 \\ -0.16590 & 0.32980 & 0 \end{pmatrix}$$

After being obtained the value of the USVT matrix, the next step is to reduce the rank of the matrix. This was done in order to reduce computing time. It is an example of a rank reduction of k = 2 from the USVT matrix as follows:

$$Uk = \begin{pmatrix} 0.04335 & 0.14351 \\ 0.04335 & 0.14351 \\ -0.12615 & 0.47331 \\ -0.12615 & 0.47331 \\ -0.12615 & 0.47331 \\ 0.15389 & 0.26501 \\ 0.04335 & 0.14351 \\ -0.12615 & 0.47331 \\ -0.11054 & -0.12150 \\ -0.16590 & 0.32980 \\ -0.16590 & 0.32980 \\ -0.16590 & 0.32980 \\ -0.16590 & 0.32980 \end{pmatrix}$$

$$Sk = \begin{pmatrix} 4.1717 & 0 \\ 0 & 2.14403 \end{pmatrix}; \; Sk\text{-}1 = \begin{pmatrix} 0.23971 & 0 \\ 0 & 0.46641 \end{pmatrix}$$

$$Vk = \begin{pmatrix} 0.64197 & 0.56819 \\ -0.46114 & -0.26051 \\ -0.70711 & 0.70711 \end{pmatrix}; \; VkT = \begin{pmatrix} 0.64197 & -0.46114 & -0.70711 \\ 0.5681 & -0.26051 & 0.70711 \end{pmatrix}$$

The last step is to calculate angle cosine value between document vactor (A) and input vactor (B) as follows:

$$Di = DiT \; Uk \; Sk\text{-}1$$

$$Di = DiT = \begin{pmatrix} 0.04335 & 0.14351 \\ 0.04335 & 0.14351 \\ -0.12615 & 0.47331 \\ -0.12615 & 0.47331 \\ -0.12615 & 0.47331 \\ 0.15389 & 0.26501 \\ 0.04335 & 0.14351 \\ -0.12615 & 0.47331 \\ -0.11054 & -0.12150 \\ -0.16590 & 0.32980 \\ -0.16590 & 0.32980 \\ -0.16590 & 0.32980 \\ -0.16590 & 0.32980 \end{pmatrix} \begin{pmatrix} 0.23971 & 0 \\ 0 & 0.46641 \end{pmatrix}$$

$$DM = (-0.03970 \quad 0.57538)$$

$$D1 = (0.64197 \quad 0.56819)$$

$$D2 = (-0.46114 \quad -0.26051)$$

$$D3 = (-0.70711 \quad 0.70711)$$

$$\text{Cos } \alpha = \frac{A.B}{|A|.|B|}$$

$$\text{Cos } \alpha_1 = \frac{(-0.03970)(0.64197) + (0.57538)(0.56819)}{\sqrt{(-0.03970)^2 + (0.57538)^2}\sqrt{(0.64197)^2 + (0.56819)^2}}$$

$$\text{Cos } \alpha_2 = \frac{(-0.03970)(-0.46114) + (0.57538)(-0.26051)}{\sqrt{(-0.03970)^2 + (0.57538)^2}\sqrt{(-0.46114)^2 + (-0.26051)^2}}$$

$$\text{Cos } \alpha_1 = 0.71113$$

$$\text{Cos } \alpha_2 = 0.43739$$

$$\text{Cos } \alpha_3 = 0.70542$$

From the results of the above calculation, it can be concluded that the arangement of documents that have the closest similarity with the input documents is document 1, document 3, and document 2.

3.3. Experiment and result evaluation

Testing is caried out by trying all the hadith queries on the system. Recall and precision values are searched by using formulas (6) and (7) [38, 39].

$$R = \frac{Number\ of\ relevan\ items\ retrieved}{Total\ number\ of\ relevan\ items\ in\ collection} \quad (6)$$

$$P = \frac{Number\ of\ relevan\ items\ retrieved}{Total\ number\ of\ items\ retrieved} \quad (7)$$

where, R is Recall, so the R value is obtained by comparing the Number of relevant items retrieved with the total number of relevant items in the collection. Recall is a document that is called from the system based on the user requests that follow the pattern of the system. The greater Recall value cannot be said as a good system or not. And, P is precision. So, the P value is obtained by comparing the number of relevant items retrieved with the Total number of items retrieved. Precision is the number of documents that are called from the relevant database after being assessed by the user with needed information. The greater the value of a system precision, the system can be said well.

The purpose of the recall and precision test is to obtain information on the search results obtained by the system. Search results can be judged by its recall and precision level. Precision can be considered a measure of accuracy while recall is perfection. The value of precision is the level of accuracy between the information requested by the user and the answers given by the system. While the Recall value is the success level of the system in rediscovering information. As for the results of the recall and precision tests and the time which is spent on searching the tested hadith, it can be seen in Table 2, Figures 3 and 4.

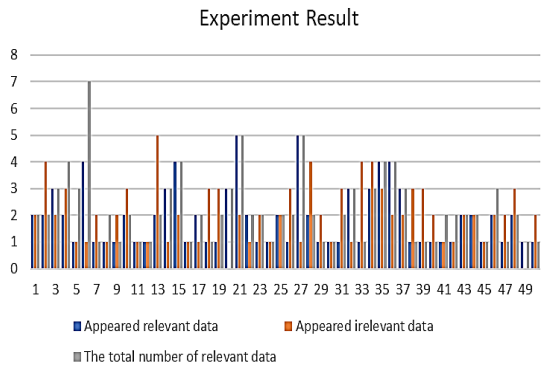

Figure 3. Result of relevant information

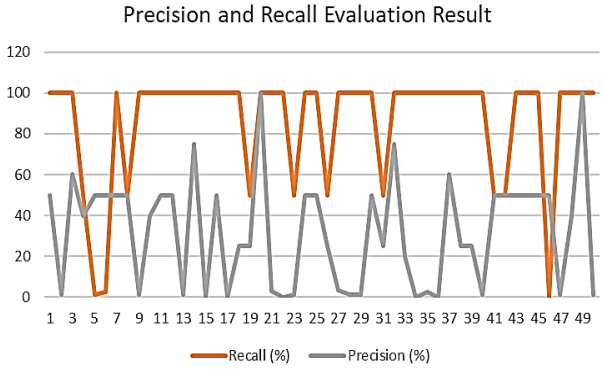

Figure 4. Result of precision and recall value

Table 2. Tested result of latent semantics analysis and cosine similarity

No	Keywords	Appeared relevant Hadith	Appeared irelevant Hadith	The total number of relevant Hadith	Recall (%)	Precision (%)
1	*Jangan berdusta atas namaku masuk neraka* (Don't lie in behalf of my name to go to the hell)	2	2	2	100	50
2	*Mendirikan shalat menunaikan zakat dan berpuasa dibulan ramadlan* (Cary out praying, alms and past in ramadan Month)	2	4	2	100	33.33
3	*Islam dibangun atas lima dasar yaitu persaksian, shalat, zakat, puasa dan ke baitullah* (Islam was formed in five pilars namely; withness, praying, alms, pasting and pilgrimage to mecca)	3	2	3	100	60
4	*Barangsiapa yang berpuasa dibulan ramadlan dengan keimanan dan ikhlas diampuni dosa-dosanya* (Whoever fasts in the month of Ramadan with faith and sincerity is forgiven of his sins)	2	3	4	50	40
5	*Malu sebagian dari iman* (Shame is part of faith)	1	1	3	33.33	50
30	*Aku pernah mandi bersama Nabi shallallahu 'alaihi wasallam dari satu bejana, dan tangan kami saling bersentuhan* (I had bathed with the Prophet sallallaahu 'alaihi wasallam from one vessel and our hands touched each other)	1	1	1	100	50
31	*Setiap Nabi memiliki doa yang dia panjatkan untuk umatnya* (Every Prophet has a prayer that he prayed for his people)	1	3	2	50	25
32	*Jika datang haid tinggalkan shalat dan bila berakhir bersikan darah lalu shalatlah* (If menstruation comes leave prayer and when it ends, clean bloody then pray)	3	1	3	100	75
33	*Tujuh puluh ribu orang dari umatku akan masuk surga, wajah mereka semua seperti rembulan* (Seventy thousand of my people will go to heaven, their faces like the moon)	1	4	1	100	20
47	*Jadikanlah (sebagian dari) shalat kalian ada di rumah kalian dan jangan jadikan kuburan* (Make (some of) your prayers in your house and do not make it a grave)	1	2	1	100	33.33
48	*Barangsiapa meninggal dalam keadaan menyekutukan Allah dengan sesuatu, maka ia masuk neraka* (Whoever dies in a state that associates God with something, he goes to hell)	2	3	2	100	40
49	*Cukuplah seseorang (dianggap) berbohong apabila dia menceritakan semua* (It is enough for someone (considered) to lie if he tells all)	1	0	1	100	100
50	*Seorang muslim yang paling baik adalah kambing yang digembalakannya di puncak gunung dan tempat-tempat terpencil* (The best Muslim is the goat that he feeds on mountain tops and remote places)	1	2	1	100	33.33
	Average (%)				87.83	36.25

4. CONCLUSION

Based on 50 times testing of the recall and precision values that have been carried out (contained in Table 2), it showed that the search engine hadith performance can apply the latent semantics analysis algorithm and cosine similarity quite well. Hadith information which is obtained based on keywords, phrases, or sentences entered successfully found well, it was indicated by a recall value of 87.83%. Although the overall information which is generated only has a value of accuracy or compliance with user input only 36.25% which is indicated by the value of the produced precision. Generally, the latent semantics analysis algorithm and cosine similarity that are used are able to produce the hadith information well. There were several factors that influenced the search results other than the possibility of an error in using the algorithm, including incomplete data and too much noise. Therefore, the pre processing stage is very important to be able to produce more accurate information. Because the pre processing stage produces text data that gives an input into the latent semantics analysis algorithm which will certainly affect the search results. For further research, the collection of saved Hadith data needs to be completed so that search engines can learn and get more precised information. In addition, the information obtained can be developed not only sorted by similarity but also can be grouped according to their meanings.

ACKNOWLEDGEMENT

Authors wishing to acknowledge Research and Publication Centre of UIN Sunan Gunung Djati Bandung that supports and funds this research publication.

REFERENCES

[1] J. M. Kassim and M. Rahmany, "Introduction to semantic search engine," *Proceedings of the 2009 International Conference on Electrical Engineering and Informatics, ICEEI 2009*, vol. 02, 2009.
[2] D. Kurniadi and A. Mulyani, "The Effect of Google's Search Engine Technology on the Development of Student Culture and Ethics (in Bahasa: Pengaruh Teknologi Mesin Pencari Google Terhadap Perkembangan Budaya dan Etika Mahasiswa)," *Jurnal Algoritma Sekolah Tinggi Teknologi Garut*, vol. 14, no. 1, 2017.
[3] P. W. Handayani, I. M. Wiryana, and J. T. Milde, "Semantic Based Search Engine For Indonesian (in Bahasa: Mesin Pencari Berbasiskan Semantik Untuk Bahasa Indonesia)," *Jurnal Sistem Informasi MTI-UII*, vol. 4, no. 2. pp. 110-114, 2012.
[4] A. Karim, "Design and Detection of the Tradition of Hadith as an Information Retrieval in the Books of Hadith (in Bahasa: Rancang Bangun Pendeteksian Keshahihan Hadits Sebagai Sebuah Information Retrieval Pada Kitab-Kitab Hadits)," *Jurnal Teknik Informatika*, vol. 5, pp. 1–20, 2012.
[5] R. N. Edi, "AS-SUNNAH (HADITS) (An Ingkar Sunnah Flow Study) (in Bahasa: AS-SUNNAH (HADITS)(Suatu Kajian Aliran Ingkar Sunnah))," *Asas*, vol. 6, no. 2, pp. 132-148, 2014.
[6] D. S. Maylawati and G. A. P. Saptawati, "Set of Frequent Word Item sets as Feature Representation for Text with Indonesian Slang," *Journal of Physics: Conference Series*, vol. 801, no. 1, pp. 1–6, 2016.
[7] H. Jiawei, M. Kamber, J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques," 3rd Edition, Elsevier, 2012.
[8] Jumadi, D. S. Maylawati, B. Subaeki, and T. Ridwan, "Opinion mining on Twitter microblogging using Support Vector Machine: Public opinion about State Islamic University of Bandung," *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*, 2016.
[9] D. S. A. Maylawati, M. A. Ramdhani, A. Rahman, and W. Darmalaksana, "Incremental technique with set of frequent word item sets for mining large Indonesian text data," *2017 5th International Conference on Cyber and IT Service Management, CITSM 2017*, 2017.
[10] A. A. Okfan Rizal Ferdiansyah, Ema Utami, "Implementation of Principal Component Analysis for Digital Image Retrieval Systems (in Bahasa: Implementasi Principal Component Analysis Untuk Sistem Temu Balik Citra Digital)," *Creative Information Technology Journal*, vol. 2, no. 3, 2015.
[11] G. Karyono, F. S. Utomo, A. Sistem, and T. Balik, "Information Retrieval in Indonesian Language Text Documents Using the Vector Space Retrieval Model Method (in Bahasa: Temu Balik Informasi Pada Dokumen Teks Berbahasa Indonesia Dengan Metode Vector Space Retrieval Model)," *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2012 (Semantik 2012)*, pp. 282–289, 2012.
[12] F. Amin, "Information Retrieval System with Vector Space Model Ranking Method (in Bahasa: Sistem Temu Kembali Informasi dengan Pemeringkatan Metode Vector Space Model)," *Dinamik*, vol. 18, no. 2, pp. 122–129, 2013.
[13] M. G. Ozsoy, F. N. Alpaslan, and I. Cicekli, "Text summarization using Latent Semantic Analysis," *Journal of Information Science*, vol. 37, no. 4, pp. 405–417, 2011.
[14] P. W. Foltz, "Latent semantic analysis for text-based research," *Behavior Research Methods*, vol. 28, no. 2, pp. 197–202, 1996.
[15] G. Cosma and M. Joy, "An Approach to Source-Code Plagiarism Detection and Investigation Using Latent Semantic Analysis," *IEEE Transactions on Computers*, vol. 61, no. 3, pp. 379–394, 2012.
[16] M. Monjurul Islam and A. S. M. Latiful Hoque, "Automated essay scoring using Generalized Latent Semantic Analysis," *2010 13th International Conference on Computer and Information Technology (ICCIT)*, 2010.

[17] T. K. Landauer, "Handbook of Latent Semantic Analysis," Psychology Press, 2014.
[18] T. K. Landauer, P. W. Folt, and D. Laham, "An introduction to latent semantic analysis," *Discourse Process.*, vol. 25, no. 2, pp. 259–284, 1998.
[19] S. T. Dumais, "Latent semantic analysis," *Annual Review of Information Science and Technology*, vol. 38, no. 1, pp. 188–230, 2005.
[20] N. E. Rozanda, A. Marsal, and K. Iswanti, "Design of the Hadith Information System Using the Vector Information Model Information Retrieval Technique (in Bahasa: Rancang Bangun Sistem Informasi Hadits Menggunakan Teknik Temu Kembali Informasi Model Ruang Vektor)," vol. 11, no. 1, pp. 1-8, 2012.
[21] M. Jamhari, E. Noersasongko, and H. Subagyo, "The Effect of Automatic Document Compacting With the Combination of Feature and Latent Semantic Analysis (LSA) Methods in the Clustering Process of Indonesian Language Text Documents (in Bahasa: Pengaruh Peringkas Dokumen Otomatis Dengan Penggabungan Metode Fitur dan Latent Semantic Analysis (LSA) Pada Proses Clustering Dokumen Teks Berbahasa Indonesia)," *Jurnal Pseudocode*, vol. 1, no. 2. pp. 72–82, 2014.
[22] P. W. Foltz, "Latent semantic analysis for text-based research," *Behavior Research Methods*, vol. 28, no. 2, pp. 197–202, 1996.
[23] M. Sofyan, "Development of Automatic Essay Correction in E-Learning at SMK PLUS AN-NABA SUKABUMI Using the Latent Semantic Analysis (LSA) Method (in Bahasa: Pengembangan Koreksi Esai Otomatis Pada E-Learning Di SMK PLUS AN-NABA SUKABUMI Dengan Menggunakan Metode Latent Semantic Analysis (LSA))," Undergraduate Theses from UNIKOM, pp. 45–52, 2015.
[24] A. Luthfiarta, J. Zeniarja, and A. Salam, "Latent Semantic Analysis (LSA) Algorithm in Automatic Document Compacting for Document Clustering Processes (in Bahasa: Algoritma Latent Semantic Analysis (LSA) Pada Peringkas Dokumen Otomatis Untuk Proses Clustering Dokumen)," *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2013 (SEMANTIK 2013)*, pp. 13–18, 2013.
[25] S. Vijayarani, J. Ilamathi, and M. Nithya, "Preprocessing Techniques for Text Mining - An Overview," *International Journal of Computer Science & Communication Networks*, vol. 5, no. 1, pp. 7–16, 2015.
[26] V. Gupta and G. S. Lehal, "A survey of text mining techniques and applications," *Journal of Emerging Technologies in Web Intelligence*, vol. 1, no. 1. pp. 60–76, 2009.
[27] K. Baker, "Singular value decomposition tutorial," Ohio State Univ., 2005.
[28] J. Ye, "Cosine similarity measures for intuitionistic fuzzy sets and their applications," *Mathematical and Computer Modelling*, vol. 53, no. 1–2, pp. 91–97, 2011.
[29] R. V. Imbar *et al*., "Implementasi Cosine Similarity dan Algoritma Smith-Waterman untuk Mendeteksi Kemiripan Teks," *Jurnal Informatika*, vol. 10, no. 1, pp. 31–42, 2014.
[30] D. S. Maylawati, W. B. Zulfikar, C. Slamet, and M. A. Ramdhani, "An Improved of Stemming Algorithm for Mining Indonesian Text with Slang on Social Media," *The 6th International Conference on Cyber and IT Service Management (CITSM 2018)*, 2018.
[31] W. Pu, N. Liu, S. Yan, J. Yan, K. Xie, and Z. Chen, "Local word bag model for text categorization," *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 625–630, 2007.
[32] A. Doucet and H. Ahonen-Myka, "An efficient any language approach for the integration of phrases in document retrieval," *Language Resources and Evaluation*, vol. 44, no. 1–2, pp. 159–180, 2010.
[33] L. Agusta, "Comparison of Porter's Stemming Algorithm with Nazief & Adriani's Algorithm for Stemming Indonesian Text Documents (in Bahasa: Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia)," *Konferensi Nasional Sistem dan Informatika 2009*, pp. 196–201, 2009.
[34] M. Adriani, J. Asian, B. Nazief, S. M. M. Tahaghoghi, and H. E. Williams, "Stemming Indonesian : A confix-stripping Approach," *ACM Transactions on Asian Language Information Processing*, vol. 6, no. 4, pp. 1–33, 2007.
[35] J. Asian, H. E. Williams, and S. M. M. Tahaghoghi, "Stemming Indonesian," *ACM Transactions on Asian Language Information Processing*, vol. 38, no. 4, pp. 307-314, 2005.
[36] A. F. Hidayatullah, C. I. Ratnasari, and S. Wisnugroho, "Analysis of Stemming Influence on Indonesian Tweet Classification," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 14, no. 4, pp. 665-673, 2016.
[37] A. S. Rizki, A. Tjahyanto, and R. Trialih, "Comparison of stemming algorithms and its effect on Indonesian text processing," T*ELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 1, pp. 95-102, 2019.
[38] L. Torgo and R. Ribeiro, "Precision and recall for regression," *International Conference on Discovery Science*, pp. 332-246, 2009.
[39] K. M. Ting, "Confusion Matrix," Encyclopedia of Machine Learning and Data Mining, Springer, 2017.