Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi

Vol. 11, No. 2, Desember 2021

ISSN 0216 – 0544
e-ISSN 2301– 6914

COMPARISON OF STEMMING AND SIMILARITY ALGORITHMS IN INDONESIAN TRANSLATED AL-QUR'AN TEXT SEARCH

[a] Ika Oktavia Suzanti, [b] Achmad Jauhari, [c] Nadila Hidayanti, [d] Iis Yuni Harianti, [e] Fifin Ayu Mufarroha

[a, b, c, d, e] Departement of Informatic Engineering, University of Trunojoyo Madura, Bangkalan – Madura, Indonesia
E-mail: iosuzanti@trunojoyo.ac.id, jauhari@trunojoyo.ac.id

Abstract

Stemming is final stage of pre-processing process in an information retrieval system. The way stemming works is to remove affixes from a word in form of prefixes, suffixes and insertions into basic word form. Selecting of best stemming algorithm is one of the ways to achieve best performance in information retrieval (IR) both in terms of speed and accuracy of search results. Thus, in this paper we did compare searches on information retrieval systems using Porter, Nazief and Adriani stemming and Enhanced Confix Stripping Stemmer with similarity methods used are cosine similarity and dice similarity. We did 8 test scenarios, first 4 scenarios are without stemming algorithm, Porter Stemming, Nazief and Adriani Stemming and Enhanced Confix Stripping Stemmer and using cosine similarity as similarity methods, while the rest of those scenario are using dice similarity. The data used is Indonesian translation of Al-quran. Based on test results, the ability to search for text is faster in stemming process with Porter Stemmer algorithm, which is 3.0438 seconds and Enhanced Confix Stripping is 3.0586 seconds. The similarity method used is cosine similarity.

Key words: Information Retrieval, Enhanced Confix Stripping, Nazief and Adriani, Cosine Similarity, Dice Similarity.

INTRODUCTION

The Holy Quran was translated in many languages, but the original text was revealed to the Prophet in Arabic language [1]. Muslims are expected to read, understand, and apply the teachings of the Holy Quran.[2]. The use of digital Qur'an has grown since the first digital copies of Qur'an in 2007 [3]. First digital version of the Qur'an was an image-based copy of the Holy book. Primarily, there are two types of digital Qur'ans: either image-based copies or text-based copies [4]. The Qur'anic text data is important, especially interpretation of Qur'an in Indonesian for most Indonesian readers [5] so that there are no errors when interpreting or understanding the meaning contained therein. Along with development of technology, many digital interpretations of Qur'an have been equipped with a search system. To make it easier for users to find translations of Qur'an interpretation according to topic, we need a search engine with concept of information retrieval (IR) [6] so that right information is obtained according to user's keywords and search results can be measured level of similarity between desired query and existing term.

The long history of information retrieval did not begin with the Internet. Prior to widespread public everyday use of search engines, information retrieval systems were found in commercial and intelligence applications since the 1960s [7]. In first half of 20th century, information retrieval was found in a variety of mobile and desktop applications [8] where the information retrieval system has increased in getting results that match the query (keyword) [9]. Search in IR can be divided into 3 kinds, namely Directed, Semidirected and undirected browsing. Directed browsing is when browsing is systematic, focused, and directed by a specific object or target. Examples include scanning a list for a known item, and verifying information such as dates or other attributes. Semidirected browsing occurs when browsing is predictive or generally purposeful: the target is less definite and browsing is less systematic. An example is entering a single, general term into a database and casually examining the retrieved records. Finally, undirected browsing occurs when there is no real goal and very little focus. Examples include flipping through a magazine and "channel-surfing." [10]. In IR there is a process to get a query or keyword through Preprocessing where the process includes tokenizing, filtering and stemming.

The approach used in stemming aims to extract the words in the user's query. The different ways of writing and using words in different sentences will give different meanings according to the grammar. Changes in the form of the words ”berhenti”, ”terhenti”, ”hentikan”, ”perhentian”, and ”menghentikan” are the development of the basic word ”henti”. The specialty of stemming refers to the heuristic method by taking the root word and cutting off the suffix. Meanwhile, data extraction with conventional methods will take time because a word-by-word search is performed on user queries. That way, data extraction on stemming not only optimizes storage, but also improves retrieval performance. As a result, stemming can improve and speed up search results significantly, which makes stemming widely used as an accuracy enhancer in IR.

Stemming is a process contained in IR that transforms words contained in a document into root words using certain rules [11]. The approach technique for Indonesian language stemming process is divided into two, with and without a dictionary. Stemming without a dictionary is Vega algorithms and algorithms Tala, while stemming the dictionary is Nazief and Adriani algorithms, algorithms and Mustapha Idris, Arifin and Setiono, confix stripping and Enhanced Confix Stripping Stemmer algorithm [12]. The porter method is a method developed for English, while confix striping stemmer, nazief stemmer, arifin stemmer, fadillah stemmer, asian stemmer, enhanced confix stripping stemmer, and arifiyanti stemmer methods were developed based on porter method for Indonesian [13].

Porter methods that are widely used in Indonesian are Porter stemmer [14], Nazief & Adriani [15], Enhanced Confix Stripping (ECS) Stemmer [16], [17][18][19]. Porter's method requires a shorter time but has a smaller percentage of accuracy (precision) compared to Nazief & Adriani [20][21]. The comparison between ECS and Porter also gives results that ECS is more accurate but the process is slow and Porter Stemmer is fastest stemming method in data processing but the results are not as accurate as ECS [22].

IR has several matching models between terms (words) and queries (keywords) that are searched for in a document: (1) Boolean model in which the document and query are represented as sets of index terms; this model is a set theoretical. (2) Probabilistic model in which the framed work for modelling document and query representation is based on probability theory; this model is probabilistic (3) Vector space model in which the documents and query are represented as vectors in t-dimensional space [23]. The vector method measures similarity between documents that will be converted into a vector model to be measured by a query. Several methods of vector space similarity include: cosine similarity and dice similarity [24]. Cosine similarity is a method to calculate level of similarity between queries and documents, to measure the similarity using cosine value function with a combination of cross product and dot product formulas from angle between two vectors [25], while dice similarity measures similarity between documents that have been retrieved using a query as a reference to calculate similarity value between documents, dice similarity is length of normalization of inner product of two vectors [26]. In this research, we will compare searches without using stemming algorithms, using Porter, Nazief & Adriani stemming and Enhanced Confix Stripping (ECS) with similarity methods used are cosine similarity and dice similarity.

MATERIAL AND METHODS

Information Retrieval is a method for retrieving information from available documents based on the query entered according to the user's wishes. Which then relevant documents is given to user [27]. Information Retrieval is an important part of searching for right information. Web search engine (Search Engine) is one of famous examples of Information Retrieval applications. Search engines mostly work by inputting queries by user, then system will search and find documents that match those query [28] . There are two stages in Information Retrieval in carrying out its main work, preprocessing and calculating similarity between documents and queries entered by user. This similarity calculation is done by applying certain methods. Preprocessing is initial process to prepare text documents to become data before further processing in next process [29]. Preprocessing includes: Case folding, tokenizing, Filtering / Stopword Removal and stemming. Case folding is the process of converting a collection of words into lowercase letters. Tokenizing is first stage in pre-processing process in preparing text documents. Tokenizing performs breakdown of a text document into a collection of words. Tokenizing is done by removing existing punctuation marks on document, then at tokenizing stage, words are separated based on spaces [13]. Filtering is a step that is carried out after tokenizing stage in pre-processing process. Filtering is a process carried out to eliminate unimportant words in text documents by checking words and comparing them whether they are included in list of unimportant words (stop list) or not [13]. If the word is included in the stop list then those words are removed. So that what is left is only words that are important which will later become keywords. Stemming is final stage of pre-processing process. Stemming cuts off affixed words into basic words by removing affixes in the form of prefixes, suffixes and insertions [11].

Stemming is final stage of pre-processing process. The way stemming works is to remove affixes from a word in form of prefixes, suffixes and insertions into basic word form [11]. The stemming algorithm for each language is different from stemming algorithm for other languages. Like English, which has a different morphology from Indonesian, stemming algorithm for the two languages is different. In English text documents, all that needs to be done is the process of decapitating the suffix. Whereas in Indonesian text documents it is more difficult because there are many types of affixes that need to be removed to get basic words of a word [21].

Algoritma Porter Stemmer

The porter stemmer algorithm developed for Indonesian is the Tala stemming algorithm developed by Fadillah Z Tala in 2003. This algorithm is an algorithm that adopts the English algorithm, namely the porter stemming algorithm developed by W.B Frakes. Stemming in this algorithm uses rule base analysis to find the basic words of a

word. The porter stemmer algorithm does not apply the use of a basic dictionary as a reference in stemming a word. In contrast to other Indonesian stemming that applies a dictionary in it such as the Nazief & Adriani, Ahmad and Vega algorithms. Because applying the rule base Porter's algorithm is an algorithm that can do stemming faster than other algorithms [28][21]. The steps of the stemming process in Porter's algorithm are:

1. Removes particle endings (-lah, -kah, -tah, -pun).
2. Deleting pronouns (Possessive Pronouns), such as (-ku, -mu, -nya).
3. Removes the first prefix. If not found, then go to step 4a, and if there is then go to step 4b.
4. a. Remove the second prefix, and continue in step 5a.
   b. Remove the suffix, if not found then the word is assumed to be the root word (root word). If found then proceed to step 5b.
5. a. Delete the suffix and the final word is assumed to be the root word (root word).
   b. Removes the second prefix and the final word is assumed to be the root word (root word). In a word, it is possible to have two consecutive prefixes.

Algoritma Nazief dan Adriani

The Nazief and Adriani algorithm is a method developed by Bobby Nazief and Mirna Adriani. This algorithm method follows the Indonesian morphology (word form) rules. The steps of the Nazief and Adriani algorithm are [21]:

- Words that have not been in stemming look at dictionary, if found, word is considered a basic right word and algorithm is stopped.
- Remove Inflectional suffixes, Derivational
- Suffixes and Derivational Prefixes. If those three things had been done but root word is not found in the dictionary, then algorithm is to restore original word before stemming.

The following are rules for using Nazief and Adriani stemming .

Table 1. Nazief and Adriani Stemming Rul**es**

Rule	Word Format	Beheading
1	berV	ber-V | be-r-V
2	berCAP	ber-CAP where C!=”r” and P!=”er”
3	berCAerV	ber-CAerV where C!=”r”
4	Belajar	bel-ajar
5	BeC1erC2	be-C1erC2 where C1!={“r” | “1”}
6	terV	ter-V | te-Rv
7	terCerV	ter-CerV where C!=”r”
8	terCP	ter-CP where C!=”r” and P!=”er”
9	teC1erC2	te-C1erC2 where C!=”r”
10	me{1|r|w|y}V	me-{1|r|w|y}V
11	mem{b|f|v}	mem-{b|f|v}
12	Mempe	mem-pe
13	mem{rV|V}	me-m{rV|V} | me-p{rV|V}
14	men{c|d|j|s|z}	men-{c|d|j|s|z}
15	menV	me-nV | me-tV
16	meng{g|h|q|k}	meng-{g|h|q|k}
17	mengV	meng-V | meng-kV
18	menyV	meny-sV
19	mempA	mem-pA where V!=”e”
20	pe{w|y}V	pe-{w|y}V
21	perV	per-V | pe-rV
22	perCAP	per-CAP where C!=”r” and P!=”er”
23	perCAerV	per-CAerV where C!=”r”
24	pem{b|f|V}	pem-{b|f|V}
25	pem{rV|V}	pe-m{rV|V} | pe-p{rV|V}
26	pen{c|d|j|z}	pen-{c|d|j|z}

Algoritma ECS Stemmer

ECS (Enhanced confix stripping) stemmer is best algorithm for stemming Indonesian language and has fewer stemming errors than previous algorithm. This algorithm is an improvement algorithm from the stemmer CS (Confix Stripping) algorithm. And the CS algorithm is a development algorithm from Nazief & Adriani algorithm by adding some

rules [30]. In ECS stemmer algorithm, there is an additional algorithm in the form of a suffix return process if the previous recoding process failed and the addition of word snippet rules as in Table 2 [31].

Table 2. Rules for word snippet ECS Stemmer [31]

Rule	Word Format	Beheading
1	berV…	ber-V
2	berCAP…	ber-CAP, where C!='r'&P!='er'
3	berCAerV	ber-CaerV... where C!="r"
4	belajar	bel-ajar
5	beC1erC2...	be-C1erC2... where C1!={"r"|"l"}
6	terV…	ter-V... | te-rV...
7	terCerV...	ter-CerV... where C!="r"
8	terCP...	ter-CP... where C!="r" and P!="er"
9	teC1erC2...	te-C1erC2... where C1!="r"
10	me{l|r|w|y}V...	me-{l|r|w|y}V...
11	mem{b|f|v}...	mem-{b|f|v}...
12	mempe...	mem-pe...
13	mem{rV|V}...	me-m{rV|V}... | me-p{rV|V}...
14	men{c|d|j|z|s}...	men-{c|d|j|z|s}...

Cosine Similarity

Cosine similarity is a method to measure the similarity between documents of two objects. The measurement uses cosine concept with a combination of the cross product and dot product formulas from the angle between the two vectors which results in the percentage of similarity values between documents and queries. The calculation result of cosine similarity produces a value of 0 to 1. 0 indicates that there is no similarity between document and the query, while the value 1 indicates the most similar value between document and query. From a distance of values from 0 to 1, they will be sorted descendingly or from the largest value to the smallest value) [25][32]. For calculation of the cosine similarity method using equation [25]:

$$Cosine\ similarity = \cos\theta = \frac{d \bullet q}{|d|\,|q|} = \frac{\sum_{i=1}^{n}(w_{di}\ x\ w_{qi})}{\sqrt{\sum_{j=1}^{n}(w_{di})^2 \cdot \sum_{j=1}^{n}(w_{qi})^2}} \quad (1)$$

where :

q = document Q to be compared with document D

d = document D to be compared with document Q

$q \bullet d$ = *dot product between vectors Q and D*

$|q|$ = length of vector q

$|d|$ = length of vector d

$|q||d|$ = *cross product between* $|q|$ *and* $|d|$

w_{di} = term weight in the i -th document, = tf x idf

w_{qi} = *term* weight in the i -th *query* , = tf x idf

i = number of terms in sentence

n = number of vectors

Dice Similarity

In this study, the Dice Similarity or Dice Coefficient method is used because it is a method to measure a similarity in the IR which is used to calculate the quantitative value (number of documents) of similarity and comparison between documents A and B. Dice similarity is the length of normalization of the inner product of two vectors for measure among documents that have been retrieved (taken) to use the query as a reference count. The resulting value of dice similarity is only 0 to 1. A value of 0 indicates that there is no similarity between document and query, while value of 1 indicates highest similarity between query and document. In this system also uses a threshold that is used as a limiting value, dice similarity result calculation will be sorted by sorting from largest to smallest value (descending) [33][34]. To measure the similarity value between documents and queries, we use the equation [34]:

$$Dice\ Similarity = (\vec{d}.q) = \frac{2|\vec{d}\,.\,\vec{q}|}{|\vec{d}|^2 + |\vec{q}|^2} = \frac{2\sum_{i=1}^{n}(P_i Q_i)}{\sum_{i=1}^{n} P_i^2 + \sum_{i=1}^{n} Q_i^2} \quad (2)$$

where :

P = document different from Q
Q = document different from P
P_i = word in document p
Q_i = word in document q

METHODOLOGY

In figure 1 is show the architecture system is made. The admin login first then inputs the query into system then system processes data through tokenizing, filtering / stopword removal, Stemming Porter, Nazief & Adriani and Enhanced Confix Stripping (ECS) stages, and TF-IDF weighting after that results are carried out in cosine calculation process similarity and dice similarity and system performs a ranking process to bring up relevant interpretation documents to admin which contains information on processing time from system when performing a search. Meanwhile, user inputs query he wants to find, then query is also processed through tokenizing, filtering / stopword removal, Porter, Nazief & Adriani and Enhanced Confix Stripping (ECS) stages, TF-IDF weighting and calculating the similarity between query and document using cosine similarity and dice similarity method. The next stage of ranking results from ranking stage produces a relevant interpretation document that is given to user. The search process is completed when system has show relevant interpretation document according to query entered by admin or user.

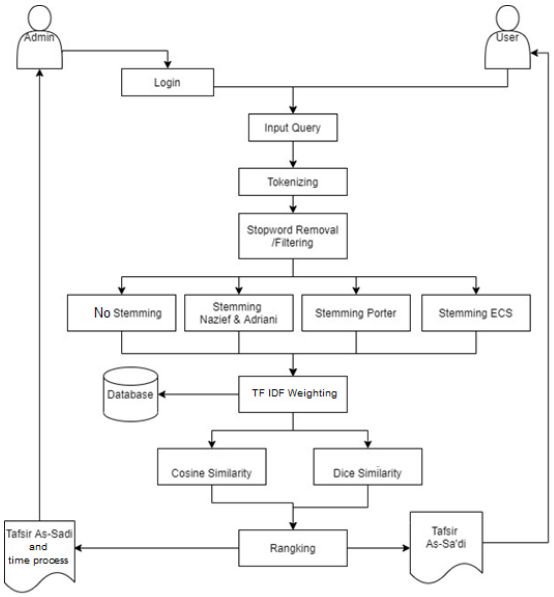

Fig 1. Architecture system

The trial in this study was conducted to determine level of accuracy generated in search process using the Stemming Porter, Nazief & Adriani algorithm and Enhanced Confix Stripping (ECS) with similarity methods, namely cosine similarity and dice similarity. To determine level of accuracy, resulting data value is then carried out by calculating percentage of F-Measure with equation [35].

$$F - Measure = 2 \frac{\text{Precision x Recall}}{\text{Precision+ Recall}} x\ 100\% \quad (3)$$

where :
Precision = precision value (level of precissioon)

Recall = recall value (ability to retrieve data)

Time trials were conducted to compare the processing time of text searches using the Stemming Porter, Nazief & Adriani algorithm and Enhanced Confix Stripping (ECS) with similarity methods, namely cosine similarity and dice similarity. Test try when done with input multiple query similar to test the calculation of Recall, Precision and F-Measure and then a process of 15 query calculated average value. With equation:

$$Avg = 2 \frac{\text{total processing time}}{\text{total query}} x\ 100\% \quad (4)$$

RESULT AND DISCUSSION

The test was conducted to find out which stemming method is better by comparing the accuracy and processing time of the use of the Stemming Porter, Nazief & Adriani and Enhanced Confix Stripping (ECS) algorithms with the similarity method, namely cosine similarity and dice similarity. Testing is done by inputting 15 different queries. The test is carried out by inputting 3 queries consisting of 1 word, 3 queries consisting of 2 words, 3 queries consisting of 3 words, 3 queries consisting of 5 words and the last 3 queries consisting of 7 words. The test is done by calculating the F-measure. The test calculation is carried out to find out what percentage of the output is in accordance with the query and what percentage of the output that appears but does not match the query from the entire existing data. Furthermore, the F-measure calculation is carried out whose calculation

results are obtained from the recall and precision results.

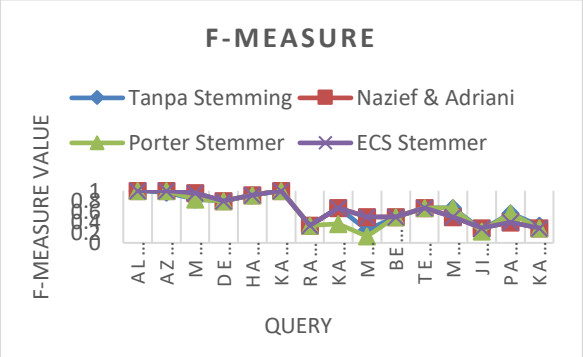

Fig 2. F-Measure for each stemming algorithm using cosine similarity

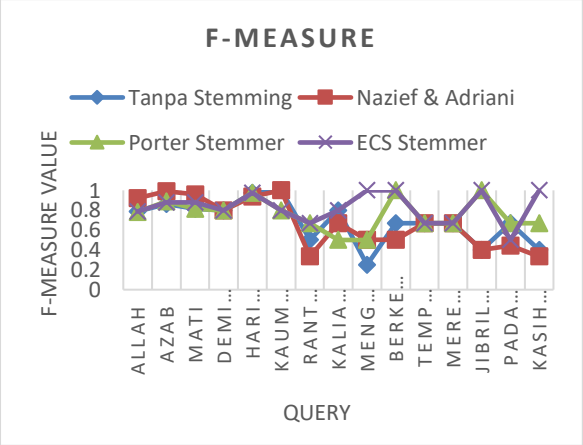

Fig 3. F-Measure for each stemming algorithm using dice similarity

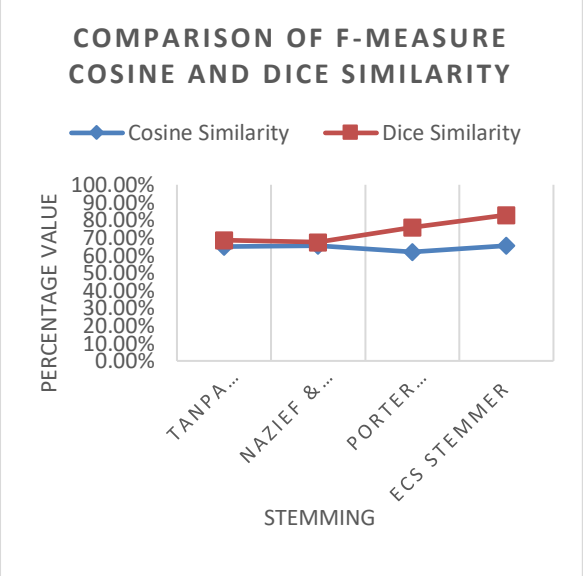

Fig 4. F-Measure for each similarity algorithm

In Figure 2 the stemming process without the Stemming algorithm, Stemming Porter, Nazief & Adriani and ECS Stemmer have varying F-Measure values. The accuracy value of all scenarios begins to experience a significant decrease when the query with the first 3 words is entered because there are documents that are not found but are relevant (TN). This happens because the similarity value of the document has a similarity value less than a predetermined threshold of 0.5. After that, the accuracy value increases again because the words entered in the query may have the same words as the query that was previously inputted. In all scenarios, the accuracy value decreases as the number of words in the entered query increases. This happens because each of these words can be translated into different basic words for each of the algorithms used so that more and more documents are retrieved. In Figure 3 the stemming process without the Stemming algorithm, Stemming Porter, Nazief & Adriani and ECS Stemmer uses dice similarity. Just like in the test scenario with cosine similarity, the accuracy value starts to decrease when a query with 3 words is entered. However, this decrease is not as much as in the cosine similiarity of 50%. In all scenarios, the accuracy value of the ECS stemmer has a significant difference with Nazief & Adriani. This happens that the number of rules in the two algorithms also has a different number of up to two times. In Figure 4, in general, the accuracy values of each similarity algorithm are compared.

The time trial was conducted to compare the processing time of the text search using the Stemming Porter, Nazief & Adriani, ECS algorithm and without the stemming algorithm with the similarity method, namely cosine similarity and dice similarity. The time trial was carried out by inputting 15 queries which were the same as the F-Measure calculation test then the processing time of the 15 queries was calculated the average value It can be seen that the accuracy value with dice similarity is higher than cosine similarity.

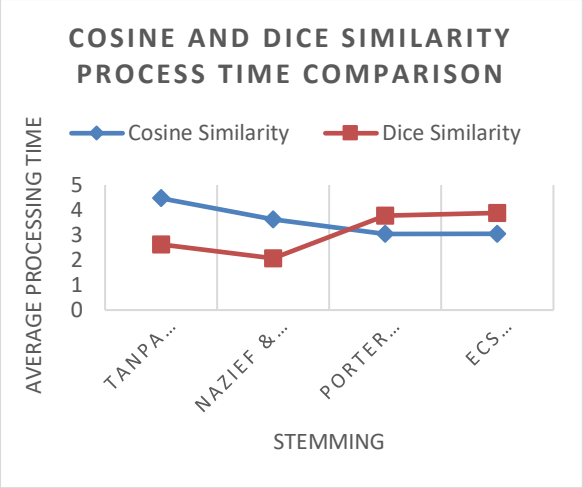

Fig 5. Time comparison for each stemming algorithm using dice similarity

CONCLUSION

Based on the text search test on the interpretation of the Qur'an from the calculation results, the accuracy of Dice similarity is higher than cosine similarity. In the process without stemming algorithm, obtained an accuracy value of 68.4% for dice similarity and 65.02% for cosine similarity. In the stemming process using the Nazief & Adriani algorithm, the accuracy values were 67.45% and 65.41%, respectively. In the stemming process using the Porter Stemmer algorithm, the accuracy values are 75.95% and 61.87%, respectively. In the stemming process using the Porter Stemmer algorithm, the accuracy values are 65.41% and 82.85%, respectively. The ability to search text on dice similarity is faster in the stemming process using the Porter Stemmer and ECS algorithms. Meanwhile, in the Nazief & Adriani algorithm and without stemming, cosine similarity is faster than dice similarity.

REFERENCES

[1] F. Malik, "The Qur'an in English Translation Complete," *Mideast. Coexistence*, 2007.

[2] A. M. Abualkishik, K. Omar, and G. A. Odiebat, "QEFSM model and Markov Algorithm for translating Quran reciting rules into Braille code," *J. King Saud Univ. Inf. Sci.*, vol. 27, no. 3, pp. 238–247, 2015.

[3] M. F. Hilmi, M. F. Haron, O. Majid, and Y. Mustapha, "Authentication of electronic version of the Holy Quran: an information security perspective," in *2013 Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences*, 2013, pp. 61–65.

[4] M. Almazrooie, A. Samsudin, A. A.-A. Gutub, M. S. Salleh, M. A. Omar, and S. A. Hassan, "Integrity verification for digital Holy Quran verses using cryptographic hash function and compression," *J. King Saud Univ. Inf. Sci.*, vol. 32, no. 1, pp. 24–34, 2020.

[5] S. Raharjo, R. Wardoyo, and A. E. Putra, "Detecting proper nouns in indonesian-language translation of the quran using a guided method," *J. King Saud Univ. Inf. Sci.*, vol. 32, no. 5, pp. 583–591, 2020.

[6] Y. S. Yogi Suntono, "Implementasi Text Mining Pada Aplikasi Search Engine Tafsir Al-Qur'an Menggunakan Metode Cosine Similarity." TEKNIK INFORMATIKA, 2017.

[7] M. Sanderson and W. B. Croft, "The history of information retrieval research," *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1444–1451, 2012.

[8] M. A. Hearst, "'Natural'search user interfaces," *Commun. ACM*, vol. 54, no. 11, pp. 60–67, 2011.

[9] P. Seethalaksmi, "Semantic search based efficient retrieval of educational multimedia information using service oriented architecture."

[10] C. W. Choo, B. Detlor, and D. Turnbull, "Information Seeking on the Web--An Integrated Model of Browsing and Searching.," 1999.

[11] A. A. Magriyanti, "Analisis Pengembangan Algoritma Porter Stemming Dalam Bahasa Indonesia," 2018.

[12] B. C. Ningrum, "Perbandingan Algoritma Stemming untuk Bahasa Indonesia dengan Parameter Akurasi dan Waktu Proses," 2019.

[13] R. Melita, "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)," Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta, 2018.

[14] R. C. N. Santi, S. Eniyati, R. Retnowati, and H. Yulianton, "PENGGUNAAN SISTEM TEMU KEMBALI DALAM PENCARIAN KATA UNTUK TERJEMAHAN AL QURAN," 2019.

[15] B. Poernomo *et al.*, "Sistem Information Retrieval Pencarian Kesamaan Ayat Terjemahan Al Quran Berbahasa Indonesia," *Semin. Nas. Teknol. Inf. dan Komun.*, pp. 100–108, 2015.

[16] I. Humaini, T. Yusnitasari, L. Wulandari, D. Ikasari, and H. Dutt, "Informatian Retrieval of Indonesian Translated version of Al Quran and Hadith Bukhori Muslim," in *2018 International Conference on Sustainable Energy, Electronics, and Computing Systems (SEEMS)*, 2018, pp. 1–5.

[17] I. Z. Amalia, A. N. P. Bimantoro, A. Z. Arifin, M. Faisol, R. Indraswari, and R. W. Sholikah, "INDONESIAN-TRANSLATED HADITH CONTENT WEIGHTING IN PSEUDO-RELEVANCE FEEDBACK QUERY EXPANSION," *J. Ilm. Kursor*, vol. 11, no. 1, 2021.

[18] W. L. Ningrum and I. Humaini, "PRE-PROCESSING PENDUKUNG INFORMATION RETRIEVAL MELALUI PEMBENTUKAN KORPUS AL-QURAN TERJEMAHAN BAHASA INDONESIA," in *SNIA (Seminar Nasional Informatika dan Aplikasinya)*, 2020, vol. 4, pp. B34-36.

[19] A. Jauhari, I. O. Suzanti, Y. D. Pramudita, and N. P. W. Diantisari, "Enhanced Confix Stripping Stemmer And Cosine Similarity For Search Engine in The Holy Qur'an Translation," in *2020 6th Information Technology International Seminar (ITIS)*, 2020, pp. 207–212.

[20] L. Agusta, "Perbandingan algoritma stemming Porter dengan algoritma Nazief & Adriani untuk stemming dokumen teks bahasa indonesia," *Konf. Nas. Sist. dan Inform.*, vol. 2009, pp. 196–201, 2009.

[21] D. Wahyudi, T. Susyanto, and D. Nugroho, "Implementasi dan analisis algoritma stemming nazief & adriani dan porter pada dokumen berbahasa indonesia," *J. Ilm. SINUS*, vol. 15, no. 2, pp. 49–56, 2017.

[22] M. Alif, F. Solihin, and H. Husni, "Perbandingan Metode Enhanced Confix Stripping dan Porter Stemmer Untuk Stemming Konten Bahasa Indonesia," 2014.

[23] R. Premalatha and S. Srinivasan, "Text processing in information retrieval system using vector space model," in *International Conference on Information Communication and Embedded Systems (ICICES2014)*, 2014, pp. 1–6.

[24] A. Jain, A. Jain, N. Chauhan, V. Singh, and N. Thakur, "Information retrieval using cosine and jaccard similarity measures in vector space model," *Int. J. Comput. Appl.*, vol. 164, no. 6, pp. 28–30, 2017.

[25] O. Nurdiana, J. Jumadi, and D. Nursantika, "Perbandingan metode Cosine Similarity dengan metode Jaccard Similarity pada aplikasi pencarian terjemah Al-Qur'an dalam Bahasa Indonesia," *J. Online Inform.*, vol. 1, no. 1, pp. 59–63, 2016.

[26] M. Chahal, "Information Retrieval using Dice Similarity Coefficient," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 6, no. 6, pp. 72–75, 2016.

[27] T. Yusnitasari, I. Humaini, L. Wulandari, and D. Ikasari, "Informatian Retrieval for Popular Words in Bahasa Translation of Al Quran and Hadith Bukhori Using Enhance Confix Stripping (ECS) Stemming," *Am. J. Softw. Eng. Appl.*, vol. 8, no. 1, p. 18, 2019.

[28] N. J. M. Verdaningroem and A. Saifudin, "Penerapan Kamus Dasar Pada Algoritma Porter Untuk Mengurangi Kesalahan Stemming Bahasa Indonesia," *J. Teknol.*, vol. 10, no. 2, pp. 103–112, 2018.

[29] M. D. R. Wahyudi, "Penerapan Algoritma Cosine Similarity pada Text

Mining Terjemah Al-Qur'an Berdasarkan Keterkaitan Topik," *Semesta Tek.*, vol. 22, no. 1, pp. 41–50, 2019.

[30] M. N. Khidfi, I. Isnawaty, and J. Y. Sari, "RANCANG BANGUN APLIKASI PENDETEKSIAN KESAMAAN PADA DOKUMEN TEKS MENGGUNAKAN ALGORITMA ENHANCED CONFIX STRIPPING DAN ALGORITMA WINNOWING," *semanTIK*, vol. 4, no. 2, pp. 1–10, 2018.

[31] Y. N. Fadziah and E. F. Rahman, "Penerapan Algoritma Enchanced Confix Stripping dalam Pengukuran Keterbacaan Teks Menggunakan Gunning Fog Index," *JATIKOM J. Apl. dan Teor. Ilmu Komput.*, vol. 1, no. 1, pp. 15–24, 2018.

[32] R. T. Wahyuni, D. Prastiyanto, and E. Supraptono, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," *J. Tek. Elektro*, vol. 9, no. 1, pp. 18–23, 2017, doi: 10.15294/jte.v9i1.10955.

[33] W. B. Croft, D. Metzler, and T. Strohman, *Search engines: Information retrieval in practice*, vol. 520. Addison-Wesley Reading, 2010.

[34] A. D. Fikri, "Perbandingan metode dice similarity dengan cosine similarity menggunakan query expansion pada pencarian ayatul ahkam dalam terjemah Alquran berbahasa Indonesia." Universitas Islam Negeri Maulana Malik Ibrahim, 2018.

[35] D. Marutho, "PERBANDINGAN METODE NAÏVE BAYES, KNN, DECISION TREE PADA LAPORAN WATER LEVEL JAKARTA," *INFOKAM*, vol. 15, no. 2, 2019.

.