Hybrid Spelling Correction and Query Expansion for Relevance Document Searching

Dewi Soyusiawaty, Denny Hilmawan Rahmatullah Wolley
Informatics Engineering, Ahmad Dahlan University, Yogyakarta, Indonesia

***Abstract*—A digital library is a type of information retrieval (IR) system. The existing IR methodologies generally have problems on keyword searching. Some of search engine has not been able to provide search results with partial matching and typographical error. Therefore, it is required to be able to provide search results that are relevant to keywords provided by the user. We proposed a model to solve the problem by combining the spell correction and query expansion. Searching is starting with indexing the title of the document by preprocessing the title of all incoming document data and then weighting the Term Frequency – Inverse Document Frequency (TF-IDF) against all terms of the whole document. Levenshtein Distance algorithm is used in the search process to correct typo-indicated keywords. Before calculating the relevance between the keywords and the documents using Cosine Similarity, the keywords are expanded using Query Expansion to increase number of documents retrieved. Calculation results using Cosine Similarity are then added to Query Expansion weight calculation to get final ranking result. Results show improvements over IR system compared with system without spell check and query expansion. The results of the study in the form of web-based application conducted testing for 50 times with number of data of 2,045. The system was able to correct typo-indicated keywords and search documents with average recall value of 95.91%, average precision value of 63.82% and average Non Interpolated Average Precision (NIAP) value of 86.29%.**

Keywords*—Cosine similarity; information retrieval; Levenshtein distance; TF-IDF; typographical error; query expansion

I. Introduction

Digital library is one of information service providers in the forms of digital documents that can be accessed online. It is very helpful for students in searching information to complete assignments as well as searching supporting documents for research they are conducting. The large number of digital documents at digilib makes the scope of information search even greater so that information and the needs of relevant information are increasing [1]. Based on observation, some digilib has not been able to provide search results with partial matching and typo-indicated keywords (typing error).

In searching information, the model used and the choice of keywords can influence the level of document relevance towards user's keywords. One of them is VSM (Vector Space Model) where this model represents documents into the forms of vector space. VSM enables to determine relevant documents with keywords depending on the similarity measurement [2]. One of popular VSM measurement models is Cosine Similarity that calculates the cosine angle between two vectors. In addition, Cosine Similarity can be implemented on document matching and partial matching [3]. With the ever increasing size of the web, relevant information extraction on the Internet with a query formed by a few keywords has become a big challenge. Query Expansion (QE) plays a crucial role in improving searches on the Internet [4]. Query expansion plays a major role in reformulating a user's initial query to a one more pertinent to the user's intended meaning. It is to retrieve the most relevant expansion words for expanding the initial query of the user in order to enhance the outcomes of web search results. The reformulated query is then used to obtain more appropriate outcomes from a large amount of information on the web [5].

Spelling errors are words that spell-checker could not find in its lexicon [6]. Typo on keywords is one of reasons why the search result is not relevant since keywords entered are not in the database, so the search engine cannot find relevant documents with the typo-indicated keywords. Several research on search engine concluded that the keyword spelling errors by users are relatively high [7]. The causes for errors are usually related to writing ignorance, positions of keyboard buttons, and finger's movement [8]. Therefore, it needs spelling correction. Levenshtein Distance, also called edit-distance, is used to find word candidates suggested based on number of minimum characters that need to be substituted, inserted, or deleted to change words from string A to be string B [9]. Levenshtein Distance provides good results in solving problems of matching string data to provide text suggestion, for instance in handwriting recognition, search words and misspelled words, so the input effectiveness increases, misspelling can be avoided and auto-complete accelerates human computer interaction [10] [11].

Based on problem explanation, the researchers conducted research to improve the relevance of document search by using Query Expansion, Levenshtein Distance algorithm and Cosine Similarity calculation.

II. Literature Review

A. Text Preprocessing

Text Preprocessing is a process to bring unstructured data form into structured one as needed for further processing in text mining. In this research, text preprocessing is for the titles of research documents and query from users [12]. It uses several general processes such as:

1) Case folding: a process of changing letters in a document into upper-case or lower-case. In this research, lower-case is used.

2) Tokenizing: a process of breaking down string into some smaller units called term. Token can be in the forms of a word / number, sentence or paragraph. In this research, term from the tokenizing result is in the form of a word [13].

3) Filtering: a process of removing symbols from string. In this research, all symbols except for alphanumeric are removed.

4) Stopword removal: a process of removing unessential words in the description by checking words of description parsing result whether they are in the unessential word list (stoplist) or not for instances are conjunction "adalah", "dan", "dari", "yang", "di" and "ke" [14].

5) Stemming: a process of removing affixes including prefixes, infixes, and or suffixes on the word group to process.

This research adds one more process to change acronym into its original form in order that term table for TF-IDF weighting becomes more structured.

B. Synonym Table Formation

Every word included in the wordlist table is then processed to find its synonym as Query Expansion reference. The search of synonym uses scraping technique towards webpage of kamuslengkap.com. The results of synonym are then stored in the wordlist table as shown in Fig. 1.

C. Query Expansion

Query Expansion reformulates user's original query to improve the effectiveness of information retrieval [15]. The use of query expansion in this research aims at increasing recall value by taking documents that have similar meaning or synonym with terms from query. The process of query expansion is done to term that has not experienced stemming [16]. Table I used for searching synonym of keywords is wordlist table containing word group from documents and the synonyms that have been found through prior scraping. For instance, if we need to search with keywords "Pencarian dokumen" (document search), each term will be expanded based on the synonym of each term in the wordlist table.

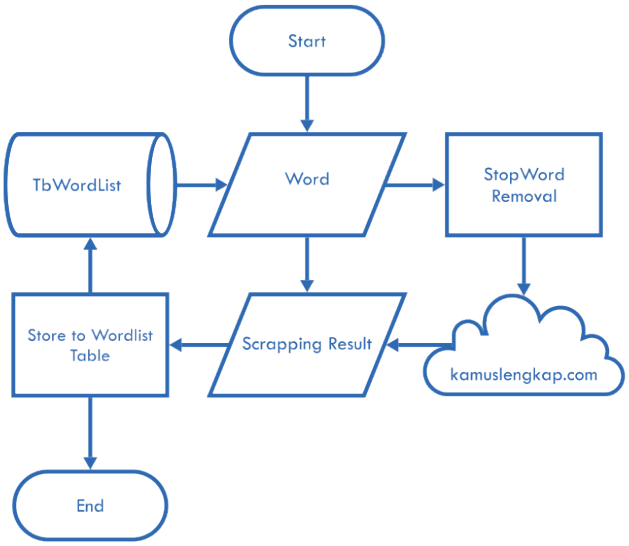

Fig. 1. Process of Synonym Table Formation.

TABLE I. SYNONYM OF KEYWORDS

Keyword	Expansion
Pencarian	*Penelusuran, Pelacakan*
Dokumen	*Arsip, Naskah*

Therefore, documents to search are not only those with terms "pencarian" and "dokumen" but also "penelusuran", "pelacakan", "arsip" and "naskah".

D. Term Frequency- Inverse Document Frequency (TF-IDF)

TF-IDF is a process of weighting a term of a document towards the whole document. TF-IDF calculation is a technique that weighs relevance of term towards document by combining two concepts in weighting, namely the frequency of a term occurs in a document (term frequency) and inverse document frequency containing the term [17].

Term Frequency (TF) that frequently occurs in documents becomes more critical since it can indicate the topic of documents. There are some formulas used to calculate term frequency in documents, but in this research TF calculation uses binary TF that focuses on a term in documents. If it is found, it scores one (1) regardless of the occurrence frequency in documents. If it is not found, it scores zero (0). Inverse Document Frequency (IDF) is used to indicate discriminative power of term i. Generally, terms that occur in a variety of documents less indicates of a particular topic. Formula of inverse document frequency is defined as followings:

$$idf_i = log\left(\frac{n+1}{df_i+1}\right) + 1 \quad (1)$$

Where df_i is document frequency of term i or number of documents containing term i and n is the number of all documents.

Weight W_ij is the multiplication result of term frequency matrix and IDF value of each term that can be defined as followings:

$$W_{ij} = tf_{ij} \times idf_i \quad (2)$$

Where df_i is document frequency of term i or number of documents containing term i and n is the number of all documents.

TF-IDF weighting is done to terms previously collected in TermList table. Fig. 2 shows the flow [18][19].

E. Spell Checker

Spellcheck is a technique that identifies incorrect words or misspelled words then changes them into correct word combinations properly. There are two main methods used to develop spelling checker application, namely identification (error detection) and correction (error correction). Besides, spelling checker is divided into two types, namely non-word error spell checker and real-word spell checker. Non-word error spell checker manages misspelled words due to typing errors, while real-word error spell checker manages substitute words to replace errors in the sentence [20] [21].

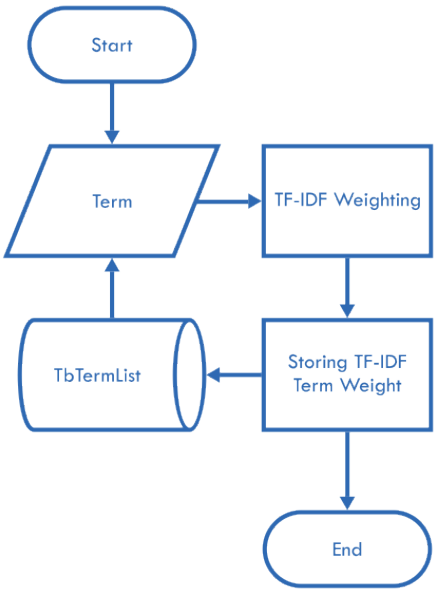

Fig. 2. Process of TF-IDF Weighting.

F. Levenshtein Distance

Levenshtein Distance is an algorithm that measures distance between two strings by calculating number of minimum operations needed to change one string to another. The operations are deletion, insertion, and substitution. Mathematically, Levenshtein Distance between two strings can be formulated as followings:

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) \\ min\begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{a_i \neq b_j} \end{cases} \end{cases} \quad (3)$$

In this research, Levenshtein Distance algorithm is used to process user's query to find out whether query typed by users is indicated typo or not. The typo-indicated words are those that are not in the wordlist table [22]. This process calculates the distance of words in query with word groups in the wordlist table. The words chosen as the correction result from the typo-indicated query words are those with the closest distance based on Levenshtein Distance calculation.

G. Relevance Calculation with Cosine Similarity

In this stage, calculation of relevance using Cosine Similarity between query and document from TF-IDF weighting previously obtained is done. The results are in the forms of similarity value between a document and the query, the higher the similarity value of a document with the query, the more relevant the document to the query. Calculating similarity between documents and query is done by dividing dot product of document vector and query vector with multiplication of Euclidean value of document vector and Euclidean value of query vector. Euclidean value is calculated by finding out the square root of the sum of the squared term weight in documents. The calculation is as the followings (4).

$$sim(d_i,q) = \frac{\vec{v}(d_i)\cdot\vec{v}(q)}{|\vec{v}(d_i)||\vec{v}(q)|}, |\vec{v}| = \sqrt{\sum_{i=1}^{M}\vec{v}_i^2(d)} \quad (4)$$

H. Calculation of Term Weight of Expansion Result

This stage adds the calculation by using IDF value of each document term of search result since the term from expansion result of query has lower degree of importance compared to that from original query. Documents containing term from original query can have higher similarity compared to those with term from expansion result. This research used calculation based on the reference as shown in (5) [23] [24].

$$sim = Cosim + \sum_{i=1}^{n}\begin{cases} 1 & ,Term = QA \\ 1-\left(\frac{1}{log(dfi)}\right) & ,Term = QE \\ 0 & ,df(QE) \leq 10 \end{cases} \quad (5)$$

Calculation is done by adding the calculation result of Cosine Similarity (Cosim) with value determined in (5). If number of term in a document is n original query term (QA), value of one (1) is added as many as n. If there is term from expansion result (QE) and df of QE > 10, so it is added 1 – (1/(log(dfi)). If df of QE ≤ 10, so it is added 0. The example of the calculation is as followings [25]:

Query: Sistem pakar penyakit tulang

Table II consists of expansion result from the tokenization of the query.

Table III consists of the calculation result of Cosine Similarity with the document.

The next is searching terms from the title of each document that intersect with original and expansion keywords in Table IV.

The intersected term results are then calculated by using equation (5) can be seen at Table V.

TABLE II. EXPANSION RESULT

Original Keyword	Expansion Keyword
Sistem	*Acara, aturan, bentuk, cara, jalan mekanisme, metode, mode, peraturan, proses, tata, teknik, system*
Pakar	*Ahli*, expert, *juru, spesalis*
Penyakit	*Kelainan, kesulitan, masalah, problem*
Tulang	-

TABLE III. CALCULATION RESULT OF COSINE SIMILARITY

ID	Title	Cosim Value
053/INF/2009	*Sistem Pakar Untuk Mendiagnosa Penyakit Tulang*	**0.2148**
047/TINF/2012	*Pemanfaatan Multimedia Dalam Sistem Pakar Untuk Mendiagnosa Penyakit Tulang*	**0.1841**
144/INF/2006	*Sistem Pakar Berbasis Web Untuk Diagnosa Penyakit Tulang Pada Manusia Menggunakan Metode* Certainty Factor	**0.1625**

TABLE IV. TERM SEARCHING

ID	Term Preprocessing	Explanation
053/INF/ 2009	*sistem*	Original Keyword
	pakar	Original Keyword
	diagnosa	-
	sakit	Original Keyword
	tulang	Original Keyword
047/TINF/ 2012	*manfaat*	-
	multimedia	-
	sistem	Original Keyword
	pakar	Original Keyword
	diagnosa	-
	sakit	Original Keyword
	tulang	Original Keyword
144/INF/ 2006	*sistem*	Original Keyword
	pakar	Original Keyword
	basis	-
	web	-
	diagnosa	-
	sakit	Original Keyword
	tulang	Original Keyword
	manusia	-
	metode	Expansion Keyword
	certainty	-
	factor	-

TABLE V. INTERSECT TERM RESULT

ID	Term Preprocessing	DF	Score	Total
053/INF/ 2009	*sistem*	961	1	4.0000
	pakar	127	1	
	sakit	148	1	
	tulang	5	1	
047/TINF/ 2012	*sistem*	961	1	4.0000
	pakar	127	1	
	sakit	148	1	
	tulang	5	1	
144/INF/ 2006	*sistem*	961	1	4.8837
	pakar	127	1	
	sakit	148	1	
	tulang	5	1	
	metode	409	1-(1/log(409))	

Total of intersected term calculation result of each document is then summed with Cosim value of each document from Table VI.

The calculation result above is the new value of document similarity calculation.

TABLE VI. INTERSECT TERM CALCULATION RESULT

ID	Title	Final Value
053/INF /2009	*Sistem Pakar Untuk Mendiagnosa Penyakit Tulang*	0.2148 + 4.0000 = **4.2148**
047/TINF/ 2012	*Pemanfaatan Multimedia Dalam Sistem Pakar Untuk Mendiagnosa Penyakit Tulang*	0.1841 + 4.0000 = **4.1841**
144/INF/ 2006	*Sistem Pakar Berbasis Web Untuk Diagnosa Penyakit Tulang Pada Manusia Menggunakan Metode* Certainty Factor	0.1625 + 4.8837 = **4.9963**

I. Recall, Precision and Non-Interpolated Average Precision (NIAP) Testing

Effectiveness of information retrieval system is the ability measurement of system to retrieve a variety of documents from database in accordance with the user's request. There are two critical things usually used in measuring ability of information retrieval system, namely ratio or comparison of recall and precision. Recall and precision calculation are calculation done to a group of documents from search result (set based measure) in overall, so it cannot describe the performance of information retrieval system in terms of relevant document ranking [2]. NIAP is used to check the search success of software developed [26].

Testing is done for 50 times for each of those with spelling correction and query expansion and those without them for testing the relevance level of document search results retrieved by system. Formula for recall, precision and NIAP are shown in (6), (7) and (8).

$$recall = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)} \quad (6)$$

$$precision = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} \quad (7)$$

$$NIAP = \frac{1}{\#relevant\ items} \times \left(\sum_{d_i \in relevant}^{n} \frac{i}{Rank\ (d_i)}\right) \quad (8)$$

III. METHOD

This research has two critical processes, namely storage and search processes.

The storage process stores data of documents consisted of id, title, year of publication, and author into document table. Then, the title of document is extracted. Term of the document title is taken for TF-IDF weighting and a group of words from documents title is taken for synonym table formation. Fig. 3 shows flow of storage process.

Document search processes consist of correction process of keywords from user's query by using Levenshtein Distance, query expansion based on synonym table formed, relevance calculation by using Cosine Similarity, and expansion term weighting process to add to relevance value of Cosine Similarity calculation result. Fig. 4 shows flow of document search process.

A. Data Collection

Dataset used in this research is in the forms of titles in excel file format containing research document titles. Data of the scraping results consist of 2,083 records with four attributes, namely ID, title, author, and year.

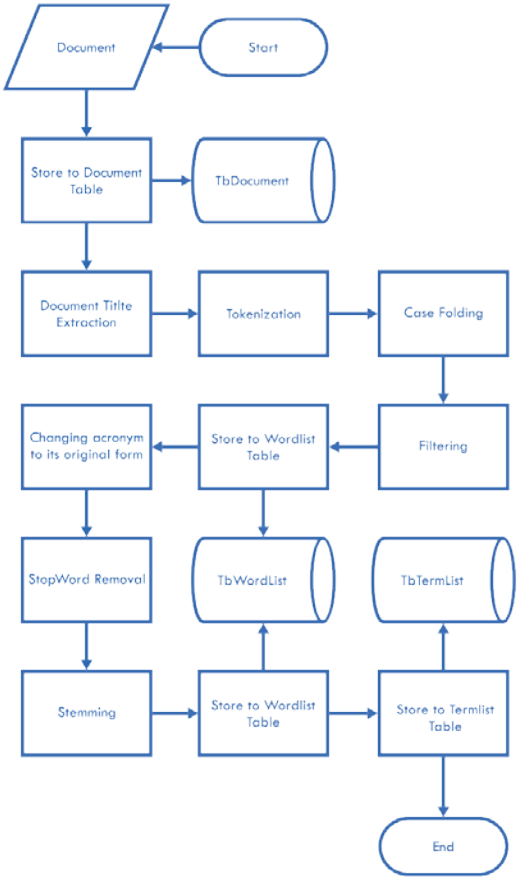

Fig. 3. Flow of Data Storage Process.

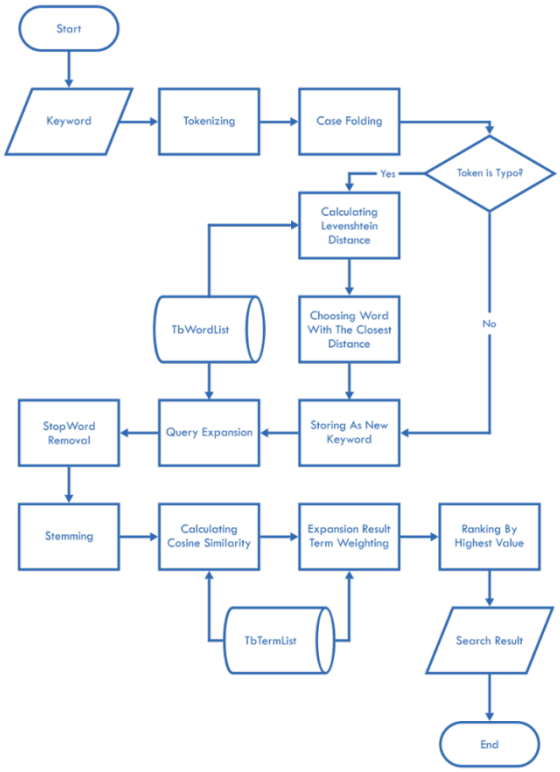

Fig. 4. Flow of Document Search Process.

IV. Result and Discussion

Testing is done for 50 times by using data of 2,045 document titles in database with different keywords in every testing both with the use of spelling correction and query expansion on query keywords or not.

A. Testing without Spelling Correction and Query Expansion

This testing only uses similarity calculation with Cosine Similarity without spelling correction and query expansion on keywords to compare recall and precision value with those with spelling correction and query expansion. In Table VII, it can be seen that several rows have errors in keyword writing which finally the Rt value = 0 and also follows the recall value, precision and NIAP = 0. Writing errors can be seen in the blocked rows, such as in the 3, 4, 5, 8, 11, 17 and so on. Various writing errors are such as missing letters, excessive letters and letter placement errors. This testing obtains average recall of 69.11%, precision of 69.28%, and NIAP of 60.72%.

TABLE VII. Searching Result without Spelling Correction and Query Expansion

No	Keyword	#Rt	Recall	Precision	NIAP
1	klasifikasi	25	48.08%	100.00%	48.08%
2	pencarian	39	78.00%	100.00%	78.00%
3	levemstein	0	0.00%	0.00%	0.00%
4	roshio	0	0.00%	0.00%	0.00%
5	steganogarfi	0	0.00%	0.00%	0.00%
6	multimedia	209	100.00%	100.00%	100.00%
7	aplikasi	161	100.00%	100.00%	100.00%
8	autocorect	0	0.00%	0.00%	0.00%
9	fuzzy	33	100.00%	100.00%	100.00%
10	bayes	61	100.00%	100.00%	100.00%
11	medfia pembelaaran	0	0.00%	0.00%	0.00%
12	klasifikasi c45	5	83.33%	100.00%	83.33%
13	teknologi informasi	22	100.00%	54.55%	61.95%
14	pengolahan citra	5	100.00%	100.00%	100.00%
15	sistem informasi	438	98.61%	97.03%	97.26%
16	web service	30	93.75%	100.00%	93.75%
17	algoritma steming	0	0.00%	0.00%	0.00%
18	kriptografi data	5	100.00%	100.00%	100.00%
19	keamanan komputer	1	100.00%	100.00%	100.00%
20	cosine similarity	0	0.00%	0.00%	0.00%
21	jaingan saraf tiruan	13	39.39%	100.00%	39.39%
22	vector space model	6	85.71%	100.00%	85.71%
23	sistem pencarian skripsi	12	100.00%	16.67%	54.17%
24	metode naive bayes	12	70.59%	100.00%	70.59%
25	lest significan bit	0	0.00%	0.00%	0.00%

26	pengembangan aplikasi mobile	16	94.12%	100.00%	94.12%
27	sistem pendukung keputusan	150	96.77%	100.00%	96.77%
28	sistem deteksi penyakit	5	7.04%	100.00%	7.04%
29	metode forwar chainng	0	0.00%	0.00%	0.00%
30	wireless application proocol	15	100.00%	100.00%	100.00%
31	sistem deteksi kemiripan dokumen	2	66.67%	100.00%	66.67%
32	perancangan sistem informasi web	2	100.00%	100.00%	100.00%
33	sistem pakar penyakit kanker	5	83.33%	100.00%	83.33%
34	metode simple additive weight	18	100.00%	94.44%	100.00%
35	sistem informasi akademik sekolah	6	60.00%	100.00%	60.00%
36	sistem deteksi penyakit tulang	9	100.00%	44.44%	95.00%
37	perancangan sistem informasi skolah	11	37.50%	27.27%	23.44%
38	media pembelajaran sekolah dasar	24	96.00%	100.00%	96.00%
39	sistem pendukung keputusan baiyes	150	100.00%	18.00%	25.31%
40	Sistem pakar metode naivebayes	49	100.00%	2.04%	3.03%
41	Pengembangan sistem berbasis web service	1	16.67%	100.00%	16.67%
42	sistem pendukung keputusan penentuan minat	1	100.00%	100.00%	100.00%
43	sistem pendukung keputusan naive bayes	1	100.00%	100.00%	100.00%
44	klasiifkasi dokumen skripsi naive baiyes	2	100.00%	100.00%	100.00%
45	analisis kinerja dosen naive bayes	1	100.00%	100.00%	100.00%
46	sistem pakar metode teorema bayes	4	50.00%	100.00%	50.00%
47	visualisasi distribusi term model vektor	1	100.00%	100.00%	100.00%
48	sistem pendukung keputusan analitical hierarcy proces	150	100.00%	9.33%	6.62%
49	sistem pakar diagnosa penyakit metode fuzy	8	0.00%	0.00%	0.00%
50	metode simple multi attribute rating technique	2	100.00%	100.00%	100.00%
	AVERAGE		**68.11%**	**69.28%**	**60.72%**

B. Testing with Spelling Correction and Query Expansion

This testing uses similarity calculation with Cosine Similarity and spelling correction on keywords with Levenshtein Distance algorithm followed with query expansion. This testing obtains average recall of 95.91%, precision of 63.82% and NIAP of 86.29%. In Table VIII it can be seen an increase in recall value, this is due to the improvement of keywords in several tests that have writing errors. The RT value was previously 0 in Table VII changes depending on the data in the database. In the correction keyword column, the rows that are blocked are spell-checked words and produce the appropriate words so that they can be found in the database. For example: "levemstein" becomes "levenshtein", "roshio" becomes "rocchio", "steganogarfi" becomes "steganography" and so on.

TABLE VIII. SEARCHING RESULT WITH SPELLING CORRECTION AND QUERY EXPANSION

No	Correction Keyword	#Rt	Recall	Precision	NIAP
1	klasifikasi	159	100.00%	32.70%	64.96%
2	pencarian	142	100.00%	35.21%	85.50%
3	levenshtein	1	100.00%	100.00%	100.00%
4	rocchio	1	100.00%	100.00%	100.00%
5	steganografi	13	100.00%	100.00%	100.00%
6	multimedia	209	100.00%	100.00%	100.00%
7	aplikasi	497	100.00%	32.39%	87.24%
8	autocorrect	1	100.00%	100.00%	100.00%
9	fuzzy	33	100.00%	100.00%	100.00%
10	bayes	61	100.00%	100.00%	100.00%
11	media pembelajaran	195	100.00%	84.62%	100.00%
12	klasifikasi c45	12	100.00%	50.00%	100.00%
13	teknologi informasi	51	54.55%	23.53%	35.88%
14	pengolahan citra	16	100.00%	31.25%	100.00%
15	sistem informasi	750	98.61%	56.67%	85.64%
16	web service	30	93.75%	100.00%	93.75%
17	algoritma stemming	4	80.00%	100.00%	80.00%
18	kriptografi data	27	100.00%	18.52%	100.00%
19	keamanan komputer	1	100.00%	100.00%	100.00%
20	cosine similarity	14	100.00%	100.00%	100.00%
21	jaringan saraf tiruan	33	100.00%	100.00%	100.00%
22	vector space model	8	100.00%	87.50%	98.21%
23	sistem pencarian skripsi	44	100.00%	4.55%	19.61%
24	metode naive bayes	49	82.35%	28.57%	76.91%
25	least significant bit	12	92.31%	100.00%	92.31%
26	pengembangan aplikasi mobile	19	100.00%	89.47%	99.67%
27	sistem pendukung keputusan	172	100.00%	90.12%	100.00%
28	sistem deteksi penyakit	96	100.00%	73.96%	80.66%

29	metode forward chaining	20	100.00%	5.00%	50.00%
30	wireless application protocol	15	100.00%	100.00%	100.00%
31	sistem deteksi kemiripan dokumen	4	100.00%	75.00%	100.00%
32	perancangan sistem informasi web	49	100.00%	4.08%	41.67%
33	sistem pakar penyakit kanker	48	100.00%	12.50%	85.42%
34	metode simple additive weighted	19	100.00%	89.47%	100.00%
35	sistem informasi akademik sekolah	29	100.00%	34.48%	85.67%
36	sistem deteksi penyakit tulang	39	100.00%	10.26%	88.75%
37	perancangan sistem informasi sekolah	30	50.00%	13.33%	54.57%
38	media pembelajaran sekolah dasar	46	96.00%	52.17%	91.52%
39	sistem pendukung keputusan bayes	94	100.00%	28.72%	84.27%
40	sistem pakar metode naivebayes	1	100.00%	100.00%	100.00%
41	pengembangan sistem berbasis web service	10	66.67%	40.00%	41.67%
42	sistem pendukung keputusan penentuan minat	84	100.00%	1.19%	100.00%
43	sistem pendukung keputusan naive bayes	19	100.00%	5.26%	100.00%
44	klasifikasi dokumen skripsi naive bayes	2	100.00%	100.00%	100.00%
45	analisis kinerja dosen naive bayes	2	100.00%	50.00%	100.00%
46	sistem pakar metode teorema bayes	4	50.00%	100.00%	50.00%
47	visualisasi distribusi term model vektor	1	100.00%	100.00%	100.00%
48	sistem pendukung keputusan analytical	12	85.71%	100.00%	85.71%
49	hierarchy process	23	100.00%	30.43%	54.90%
50	sistem pakar diagnosa penyakit metode fuzzy	2	100.00%	100.00%	100.00%
	AVERAGE		**95.91%**	**63.82%**	**86.29%**

Search results with keyword correction have higher average recall and precision value than those without keyword correction even though each trial has same keywords. The reason is that the use of Levenshtein Distance algorithm can correct some typo-indicated keywords correctly, so that system can find documents that have the keywords.

Fig. 5, Fig. 6, and Fig. 7 show comparison of recall, precision and NIAP value based on the number of term.

Fig. 5 shows that recall value of testing with spelling correction and query expansion is higher than that without them. The reason is that the use of spelling correction can display documents with typo-indicated keywords while the use of query expansion can find documents that have similar meaning with keywords from user's query.

Fig. 6 shows that precision value of testing with spelling correction and query expansion tends to be lower than that without them. The reason is that query expansions on keywords are too general; as a result, some documents that are less relevant are also displayed.

NIAP value in Fig. 7 shows the excellence of testing with spelling correction and query expansion compared to that without them. The reason is that the use of query expansion can display relevant documents that have similar meaning with keywords from the query and the ranking result is above the documents that are less relevant based on Cosine Similarity and term weighting of query expansion.

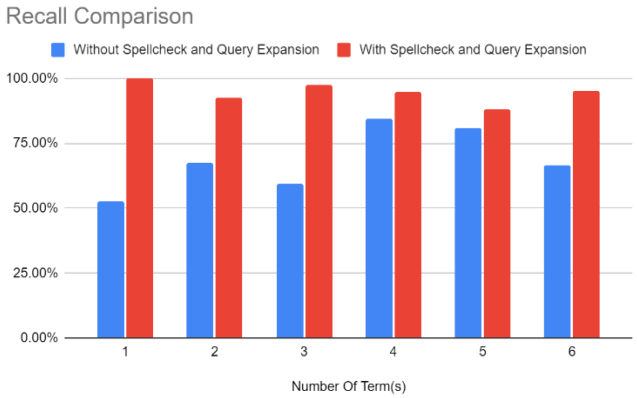

Fig. 5. Comparison of Recall Value on Number of Term.

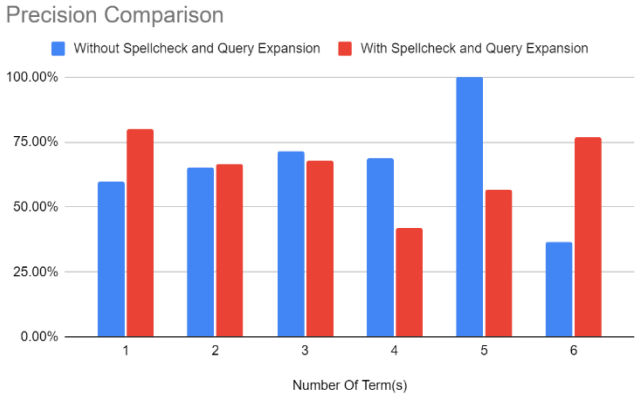

Fig. 6. Precision Comparison.

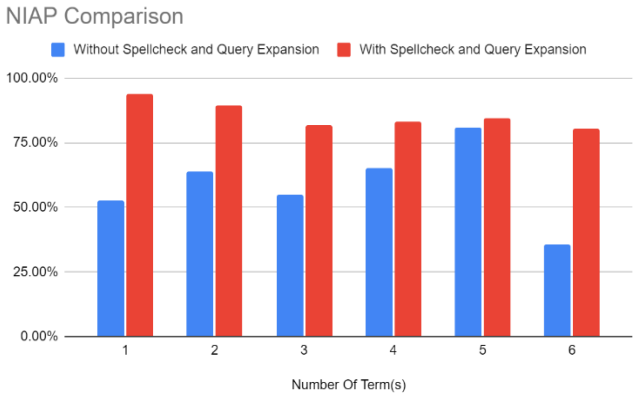

Fig. 7. NIAP Comparison.

V. Conclusion

Hybrid spelling correction and query expansion on keywords are able to improve relevance of document searching. The reason is that the use of spelling correction can display documents with typo-indicated keywords while the use of query expansion can find documents that have similar meaning with keywords from user's query. The proposed methods are able to improve relevance of document searching with average recall from 68.11% to 95.00%, but the precision decreases from 69.28% to 63.32%. However, the decrease does not influence the ranking of relevant documents retrieved by system because there is an increase of NIAP value from 60.72% to 86.29%, so the low precision is tolerable.

VI. Future Work

This research can be further developed by optimizing algorithm to correct typo-indicated keywords, so it can correct typo-indicated keywords based on the linkages of the keywords in query and not only based on their Levenshtein Distance. Besides, query expansion can be optimized in finding synonym of keywords, so term of expansion results are not too wide and to add feature to find expansion results of phrases, so it is not limited to only term from user's query. Therefore, the recall value can increase without lowering the precision value.

References

[1] Alokluk, J. A., & Al-Amri, A. (2021). Evaluation of a Digital Library: An Experimental Study. Journal of Service Science and Management, 14, 96-114.

[2] Bakala, N. "Information retrieval system by using vector space model." Int. J. of Scientific and Technol. Research 8, no. 10 (2019): 1563-8.

[3] Walia, Tarandeep Singh, Tarek Frikha, Omar Cheikhrouhou and Habib Hamam. "Comparative Study on Feature-Based Scoring Using Vector Space Modelling System. Mathematical Problems in Engineering, Volume 2021, 9946573.

[4] Azad, Hiteshwar Kumar, and Akshay Deepak. "Query expansion techniques for information retrieval: a survey." Information Processing & Management 56, no. 5 (2019): 1698-1735.

[5] Kumar Azad, Hiteshwar, Akshay Deepak, Kumar Abhishek. "Query Expansion for Improving Web Search.

[6] Hladek, Daniel, Jan Stas, Matus Pleva. "Survey of Automatic Spelling Correction". Electronics, 2020.

[7] K. T. Patil, R. P. Bhavsar and B. V. Pawar, "Word Suggestions for non-word Text Errors using Similarity Measure," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 892-897, doi: 10.1109/ICACCS51430.2021.9441858.

[8] M. Nguyen, G. H. Ngo and N. F. Chen, "Domain-Shift Conditioning Using Adaptable Filtering Via Hierarchical Embeddings for Robust Chinese Spell Check," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 2027-2036, 2021, doi: 10.1109/TASLP.2021.3083108.

[9] S. Abdulmalek, S. AL-Hagree, M. Alsurori, M. Hadwan, A. Aqlan and F. Alqasemi, "Levenshtein's Algorithm On English and Arabic: A Survey," 2021 International Conference of Technology, Science and Administration (ICTSA), 2021, pp. 1-6, doi: 10.1109/ICTSA52017.2021.9406547.

[10] Christanti M, Viny, Rudy, Dali S. Naga. "Fast and Accurate Spelling Correction using Trie and Damerau – Levenshtein Distance Bigram". Telkomnika, Vol 16, No 2, April 2018, pp 827-833.

[11] Yulianto, Muhamad Maulana, Riza Arifudin, and AlamsyahAlamsyah. "Autocomplete and spell checking Levenshtein distance algorithm to getting Text Suggest Error Data Searching in Library." Scientific Journal of Informatics 5, no. 1 (2018): 75.

[12] S. S. Pandya and N. B. Kalani, "Preprocessing Phase of Text Sequence Generation for Gujarati Language," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 749-752, doi: 10.1109/ICCMC51019.2021.9418046.

[13] Lahitani, Alfirna Rizqi, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. "Cosine similarity to determine similarity measure: Study case in online essay assessment." In 2016 4th International Conference on Cyber and IT Service Management, pp. 1-6. IEEE, 2016.

[14] D. Soyusiawaty and Y. Zakaria, "Book Data Content Similarity Detector With Cosine Similarity (Case study on digilib.uad.ac.id)," 2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA), 2018, pp. 1-6, doi: 10.1109/TSSA.2018.8708758.

[15] Afuan, Lasmedi, Ahmad Ashari, Yohanes Suyanto. "A New Approach in Query Expansion Methods for Improving Information Retrieval". Jurnal Informatika Vol 9 No 1 (2021).

[16] Afuan, Lasmedi, Ahmad Ashari, Yohanes Suyanto. "A Study : Query Expansion Method in Information Retrieval". Journal of Physics : Conference Series, Volume 1367 (2019).

[17] Rai A., Borah S. (2021) Study of Various Methods for Tokenization. In: Mandal J., Mukhopadhyay S., Roy A. (eds) Applications of Internet of Things. Lecture Notes in Networks and Systems, vol 137. Springer, Singapore.

[18] Siregar, Amril Mutoi. "Perbandingan Pembobotan Kata Dalam SistemTemu Balik Informasi." Techno Xplore: JurnalIlmuKomputer dan TeknologiInformasi 2, no. 2 (2017).

[19] M. Maryamah, A. Z. Arifin, R. Sarno and A. M. Hasan, "Adapting Google Translate using Dictionary and Word Embedding for Arabic-Indonesian Cross-lingual Information Retrieval," 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2021, pp. 205-209, doi: 10.1109/IoTaIS50849.2021.9359708.

[20] Ratnasari, C. Indah, Sri Kusumadewi, and Linda Rosita. "A Non-Word Error Spell Checker for Patient Complaints in Bahasa Indonesia." Int. J. Inf. Technol. Comput. Sci. Open Source 1, no. 1 (2017): 18-21.

[21] Maulana Yulianto, Muhammad, Riza Arifudin, Alamsyah. "Autocomplete and Spell Checking Levenshtein Distance Algorithm to Getting Text Suggest Error Data Searching in Library". Scientific Journal of Informatics, Vol. 5, No 1 (2018).

[22] D. Soyusiawaty, A. H. S. Jones and P. Widiandana, "Similarity Detection of Student Assignments Using Rocchio Method," 2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA), 2018, pp. 1-4, doi: 10.1109/TSSA.2018.8708827.

[23] Putra, Fatra Nonggala, Ari Effendi, and Agus Zainal Arifin. "Pembobotan Kata pada Query Expansion dengan Tesaurus dalam Pencarian Dokumen Bahasa Indonesia." Jurnal Linguistik Komputasional 1, no. 1 (2018): 17-22.

[24] Laxmi, Mahdarani Dwi, and Mochammad Ali Fauzi Indriati. "Query Expansion Pada SistemTemu Kembali Informasi Berbahasa Indonesia Dengan Metode Pembobotan TF-IDF Dan Algoritme Cosine Similarity Berbasis Wordnet." JurnalPengembanganTeknologiInformasi dan IlmuKomputer e-ISSN 2548 (2018): 964X.

[25] Afuan, Lasmedi, Ahmad Ashari, Yohanes Suyanto. "Query Expansion in Information Retrieval using Frequent Pattern (FP) Growth Algorithm for Frequent Item Search and Association Rules Mining". International Journal of Advanced Computer Science and Application, Vol 10, No 2 (2019).

[26] Jafar Zaidi, Syed Ali, Safdar Hussain and Samir Brahim Belhaouari. "Implementation of Text Base Information Retrieval Technique". International Journal of Advanced Computer Science and Applications, Vol. 11, No. 11 (2020).