Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi

Vol. 11, No. 1, Juli 2021

ISSN 0216 – 0544
e-ISSN 2301– 6914

INDONESIAN-TRANSLATED HADITH CONTENT WEIGHTING IN PSEUDO-RELEVANCE FEEDBACK QUERY EXPANSION

[a]Akbar Noto Ponco Bimantoro, [b]Ivanda Zevi Amalia, [c]Agus Zainal Arifin, [d]Maryamah, [e]Riska Wakhidatus Sholikah, [f]Rarasmaya Indraswari

[a, b, c, d] Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[e] Department of Information Systems, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[f] Department of Information Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
E-mail: akbar.noto16@mhs.if.its.ac.id, zevi16@mhs.if.its.ac.id, agusza@cs.its.ac.id, maryamahfaisol02@gmail.com, raras@its.ac.id, wakhidatus@its.ac.id

Abstract

In general, hadith consists of isnad and matan (content). Matan can be separated into several components for example a story, main content, and some additional information. Other texts besides main content, such as isnad and story can interfere the retrieval process of relevant documents because most users typically use simple queries. Thus, in this paper, we proposed a Named Entity Recognition (NER) component weighting model in improving the Indonesian hadith retrieval system. We did 3 test scenarios, the first scenario (S1) did not separate the hadith into several components, the second scenario (S2) separated the hadith into 2 components, isnad and matan, and the third scenario separated the hadith into 4 components, isnad, background story, content, and additional information. From the experimental results, it is found that the TF-IDF with rocchio algorithm in query expansion outperforms DocVec. Also, separation and weighting of the hadith components affect the retrieval performance because isnad can be considered as noise in a query. Separation of 2 separate components had the best overall results in general although 4 separate components showed better results in some cases with precision up to 100% and 70% recall.

Key words: Indonesian translated hadith, Information retrieval system, Named Entity Recognition, Pseudo-Relevance Feedback, Query Expansion.

INTRODUCTION

Nowadays it is easier to get literature on hadith. However, the more hadith literature, the more difficult it will be to get the hadith information needed. Currently, not many hadith collection systems have been developed. The limitations of the hadith collection system are mainly found in non-Arabic language environments. In Indonesia, the hadith retrieval system still has low-quality search results. This is because the system executes the input query directly to the DBMS [1].

Most experiments related to IRS in Hadith manuscripts usually focusing on IRS or NLP technique implementation. Amirah et al proposed a new parallel Latent Semantic Indexing (LSI) algorithm to increase the speed in finding matches with queries [2]. Meanwhile, Aulia et al built a repository and retrieval system for the collection of hadith in Indonesian to make it easier to distinguish fake hadith from authentic hadith [3]. Rasyidi et al and Rahman et al apply the thesaurus to expand the query during the search process [1], [4]. Humaini et al applying the Information Retrieval System to obtain information about the translation of the Qur'an and hadith in the Indonesian language [5].

Generally, hadith consists of *isnad* and the *matan* (content) [6]. *Isnad* is a text representing a chain transmitter of the content to ensure the validity of the hadith. The *matan* itself usually can be separated into some components e.g., conversation, story behind the hadith, the law or main content, and additional information from its transmitter. This context information can be useful in IRS since most users usually using a simple form of queries. Thus, unlikely searched information such as isnad can affect the result of the retrieval. Whereas other texts besides content, for example, *isnad*, background story, and its additional information in the hadiths can become noise that interferes with the search for hadith texts, resulting in a difficult system in retrieving relevant hadiths.

In this paper, we propose a weighting model of hadith's component-based on Named Entity Recognition (NER) in improving the Indonesian hadith retrieval system. Our proposed weighting method not only considers hadith content but also other components including *isnad*, background story, and its additional information. We separate each hadith into some components using delimiter and NER. Our hadith search system will serve users better results as user queries are generally only relevant to the content and not relevant to other components.

RELATED WORKS

Getting relevant information from a collection of documents is not easy, and nowadays there are many documents or books about hadith collections. Amirah et al aim to improve the performance of retrieval of relevant documents. The proposed solution is to use a parallel LSI algorithm. This algorithm uses a fork-join approach to perform matrix calculations automatically. The test data used came from four volumes of the Bukhari book in Malay with a total of 2028 text documents. After comparing the processing time between sequential LSI and parallel LSI, the result is parallel LSI faster than sequential LSI. Also, the results of measuring the effectiveness, precision, and recall of parallel LSIs are comparable to sequential LSIs. For more details, the average recall of both LSI systems is 49.13%. While the average precision of parallel LSI is 2.90% and sequential LSI is 2.91%. The average effectiveness of the parallel LSI was 89.76% and the sequential LSI was 89.74%. [2].

From a lot of existing hadith documents, it is not easy to distinguish which is authentic and fake hadith apart from the original source documents which are usually in Arabic. Aulia et al aim to build a repository and retrieval system for the collection of hadith in Indonesian. The Nazief and Andriani stemming algorithms are used for the document retrieval process, while the basis for the repository uses an XML schema, and search results based on keywords are displayed using the PHP programming language. The author uses a collection of hadiths from the first three books of Imam Bukhari's collection, totaling 134 hadiths. The test was conducted using 20 keywords in 2 trial scenarios. The first trial is that 20 keywords are tested without using a stemming algorithm. While the second trial of 20 keywords was tested using a stemming algorithm. The test results obtained better recall and precision using the stemming algorithm. By using the stemming algorithm, a recall value of 1 and a precision of 0.961 were

obtained, while without the stemming algorithm, a recall and precision value of 0.75 were obtained. [3].

Meanwhile, Humaini et al use the ECS (Enhance Confix Stripping) stemming method to obtain information about the translation of the hadith and the Indonesian Qur'an. The database used is a hadith document containing 7008 hadiths from Bukhari Muslim and a translation of 30 Juz Al Qur'an from the Ministry of Religion of the Republic of Indonesia. This research carried out the preprocessing stage starting from tokenization, then continued with the removal of the stopword to stemming. As mentioned earlier, the stemming process uses the ECS method. This study also built a synonym database. The synonym database was built so that information searches also pay attention to synonyms of the keywords used. So that a broader information search result is obtained. Then to obtain documents relevant to the query, the calculation of TF-IDF weights is used and the retrieval using VSM. The proposed method succeeded in accelerating the search process and obtaining relevant information [5].

Still discussing the problem of the difficulty of obtaining relevant hadith information in non-Arabic languages. Rasyidi et al and Rahman et al apply a thesaurus in the system for query expansion [1], [4]. Rasyidi et al applies a thesaurus in the hadith retrieval system in Indonesian. In research conducted by Rasyidi et al, the process of making a thesaurus used automatic co-occurrence analysis. Then proceed with the manual validation process. After conducting the trial, the accuracy results for Precision at 10 increased by 34%, the Mean Average Precision value increased by 16% and the recall value increased by 34% [1]. Rahman et al applies the thesaurus in the hadith retrieval system in Malay. This research succeeded in increasing the effectiveness of the collection by 4% [4]. So it can be concluded that both studies have proven that the use of a thesaurus has succeeded in increasing the effectiveness of obtaining relevant information.

METHODOLOGY

In general, we propose hadith component separation in retrieval system. Therefore, the system will calculate the similarities of the query with the documents based on the proposed components. Then, the system will sort the weighted mean of the similarities of each component to return *n*-most relevant documents. The proposed method applied in this study is shown in Figure 1.

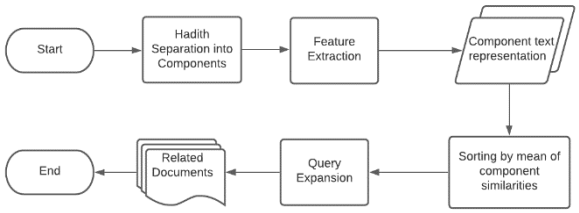

Fig 1. Proposed method.

Component Separation

In this study, we separate the hadith into *isnad*, *matan*, background story, main content, and additional information. *Isnad* is the chain of transmitter to verify the hadith validity, *matan* is the whole content of the hadith which usually contains background story or story behind why law in hadith created; main content or law that can be used as law references; and additional information such as the validity of the hadith or an information that explains there are some different pronunciation or word choices between transmitters. The overview of the relation between hadith's component and scenario are shown in Figure 2.

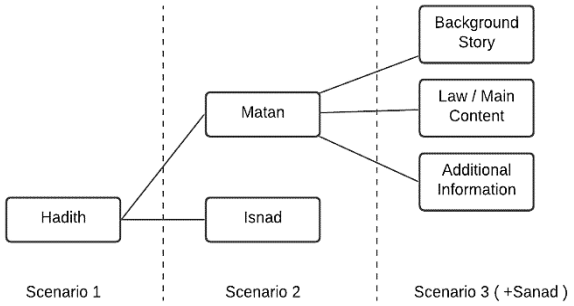

Fig 2. Component separation

We use different methods to split the hadith into components. In scenario 2, we only use delimiter or common words that separate between *isnad* and *matan* in translated hadith. The delimiter and common words are chosen from pattern observation of the dataset. This method surprisingly produced a good result because there is an explicit pattern that separating between *isnad* and *matan* such as ";" delimiter or "Rasulullah" word.

However, there is no specific pattern about how the *matan* is constructed. Some hadith contain all 3 components (background story,

main content, and additional information), but other hadith only has main content, etc. Also, usually there is no order in writing those 3 components in *matan* such as main content can be written before the background story or vice verca. These characteristics made content separation difficult. Thus, we use NER to detect the components. Each detected component will be used in Equation (1) which will be discussed later. The NER was created by manually labeling half of the datasets into several sections using Prodigy. Details on how to label NER will be explained further in the experimental results section.

Query Expansion

The next step is feature extraction of the text into values that are understood by a computer. In this study, we use the TF-IDF method. Therefore, we will use *rocchio* algorithm to optimize TF-IDF with pseudo-relevance feedback query expansion technique. We also compare the result of TF-IDF with a newer algorithm such as Doc2vec that is introduced by Let & Mikolov in 2014 [7].

Given the separated components of the hadith, the document will be sorted by weighted mean of cosine similarities that is defined in Equation (1) where *a*, *b*, *c*, and *d* are the component's weight, q_x and m_{ix} denotes the query and *i*-th document of respective *x*-component of the hadith.

The input of pseudo-relevance query expansion with *rocchio* algorithm is the most *n* and least *n* similar of the newly sorted documents. The function is defined in Equation (2) [8] where $\vec{q}$ is modified query, $\overrightarrow{q_0}$ is initial query, D_r is set of known relevant documents, D_{nr} is set of known unrelevant documents, and a, β, γ are the weight of initial query, set of known relevant documents, and set of unrelevant documents respectively.

$$\begin{aligned} weighted\ mean \\ &= a * cosim(\overrightarrow{q_a}, \overrightarrow{m_{ia}}) \\ &+ b * cosim(\overrightarrow{q_b}, \overrightarrow{m_{ib}}) \\ &+ c * cosim(\overrightarrow{q_c}, \overrightarrow{m_{ic}}) \\ &+ d * cosim(\overrightarrow{q_d}, \overrightarrow{m_{id}}) \end{aligned} \quad (1)$$

$$\vec{q} = a\overrightarrow{q_0} + \beta \frac{1}{|D_r|} \sum_{\overrightarrow{d_j} \epsilon D_r} \overrightarrow{d_j} + \gamma \frac{1}{|D_{nr}|} \sum_{\overrightarrow{d_j} \epsilon D_{nr}} \overrightarrow{d_j} \quad (2)$$

In this study, we will use 8, 16, and 4 as a, β, and γ respectively. These values are obtained from previous works by Buckley et al [9]. Also, we will use top 3 and least 3 of sorted documents as a set of known relevant documents and unrelevant documents.

EXPERIMENTAL RESULTS

Data Description

We conducted experiments on translated Abu Daud Hadith that is obtained from the previous classification study [10] which contains 4590 documents with various categories. The dataset used in this study consists of 20 to 1956 words for each hadith. More details of the dataset used in this experiment are described in Table 1.

Table 1. Details of The Dataset Used

Description	Count
Minimum word count	20
Maximum word count	1596
Mean of word count	107.28
Median of word count	82

Named Entity Recognition (NER)

To perform manual annotation on Prodigy, a dataset is needed so that the user can group the annotated results. Manual NER annotation is performed to manually label word entities that are considered important. There are 4 types of labels used, namely "SANAT", "CERITA", "INTISARI", and "KETERANGAN". Not all hadith contain these four labels. Prodigy display after annotation can be seen in Figure 3.

The results of manual annotations serve as a pre-trained model. The pre-trained model is used to enrich the dataset with automatic annotation (automatically label the text). However, the labeled entities are not used immediately, a validation process is carried out manually to justify the annotated words. After the dataset enrichment process is done, the final model is built to serve as the NER used in this proposed retrieval system.

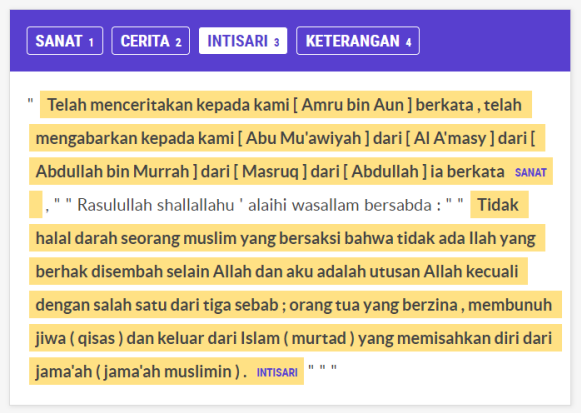

Fig 3. Example of prodigy after annotations

Usually, NER is built to detects entities which consist of some words. However, due to simplicity, we use NER to detect sentences or phrases instead of words. Nevertheless, the results of the NER evaluation were quite good, which is 72%.

Evaluation Setup

In this paper, testing the information retrieval process was carried out by two evaluation matrices namely recall and precision. Recall and precision are measurements which commonly used in information retrieval testing. Precision is the ratio of documents found and considered relevant for the searcher's needs of all documents found [11]. The formula for calculating precision [12] in information retrieval denotes in Equation (3). Meanwhile, Recall is the ratio of documents that have been found to be recovered in a search process in the information retrieval system of all relevant documents [11]. The formula for calculating recall [12] in information retrieval denotes in Equation (4).

$$Precision = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{retrieved\}|} \quad (3)$$

$$Recall = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{relevant\}|} \quad (4)$$

The precision and recall values are basically between 0-1. So that a good information retrieval system can give results close to 1 [13].

In this experiment, we use sastrawi as a stemmer which algorithm was optimized based on various researches [14–17]. Moreover, we tested the system with a simple query which keyword exists in the hadith such as "*buang air kecil*" (urinating). We also added more complicated keywords into the query such as "*bersuci setelah buang air kecil*" (wash after urinating) or "*tata cara bersuci setelah buang air kecil"* (wash after urinating procedures).

Experimental Setup

In this experiment, there are 3 scenarios conducted. The first scenario is using full text of hadith which will be called Scenaro 1 (S1). The second scenario is assigning weight into separated hadith which consist of *isnad* and *matan* (S2). The last scenario is the extension of S2 which separate *matan* into 3 more detailed component namely *background story*, *main content*, and *additional information*. In order to find which method resulting the best results between our 3 different scenarios, we change the weight of the component for S2 and S3. We can not change the weight of S1 as the data used in the IRS is not separated.

In S2, we gradually increase the *matan*'s weight (w_{matan}) from 0 to 1 by 0.1 for each iteration. Thus, *isnad*'s weight ($w_{isnad,s2}$) drops from 1 to 0 as the w_{matan} increase. These two weight values are defined in Equation (5)-(6). Meanwhile, in S3 we put different component weighting scheme for $w_{isnad,s3}$, story (w_{story}), main content (w_{main}), and additional information (w_{info}). We set a static 0.05 value for both *isnad* and additional information. As in the previous scenario (S2), the weight of the main component (w_{main}) gradually increased from 0 to 0.9. The weight values of S3 are defined in Equation (7) - (10).

$$w_{matan} = \{0, 0.1, 0.2, 0.3, \dots, 1\} \quad (5)$$

$$w_{isnad,s2} = 1 - w_{matan} \quad (6)$$

$$w_{isnad,s3} = 0.05 \quad (7)$$

$$w_{info} = 0.05 \quad (8)$$

$$w_{main} = \{0, 0.1, 0.2, 0.3, \dots, 0.9\} \quad (9)$$

$$w_{story} = 1 - w_{main} - 0.1 \quad (10)$$

The reason we choose the S2 weighting sceme is to prove that *isnad* can be a noise in vector space. Furthermore, we want to show that *matan* can be separated into pre-defined components and increase the IRS

performance. In short, we want to show that important context of the hadith need to be treated specially as people usually tends to use simple form of queries. The result of each scenario used in this paper will be discussed later in this section.

The Impact of Preprocessing

From the experiment conducted, we found that most stemmed datasets resulting a better result both precision and recall as shown in Figure 5 and Figure 6 although some scenarios show that stemming worsen the proposed method shown in Figure 4.

The experimental results show that stemming resulting a significant increase on our proposed method. As shown in Figure 5 and Figure 6, both precision and recall increased up to 10%. This shows that preprocessing used in this study, stemming, is an important step in this retrieval system. Therefore, later in this discussion, we will only present and discuss the performance of the experimental result of each scenario that is using a *stemming* method.

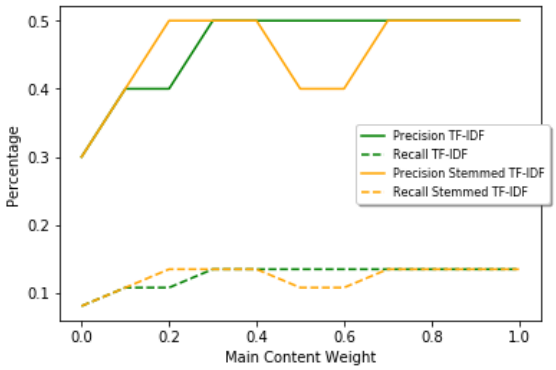

Fig 4. The effect of stemming on 10 documents retrieval.

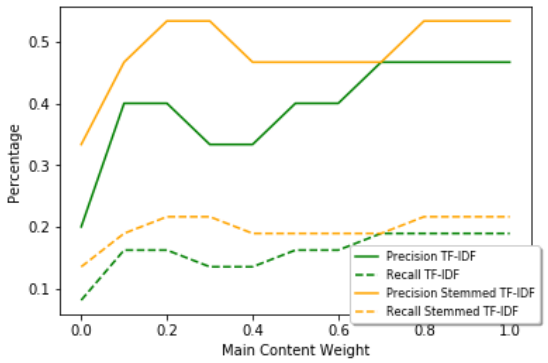

Fig 5. The effect of stemming on 15 documents retrieval.

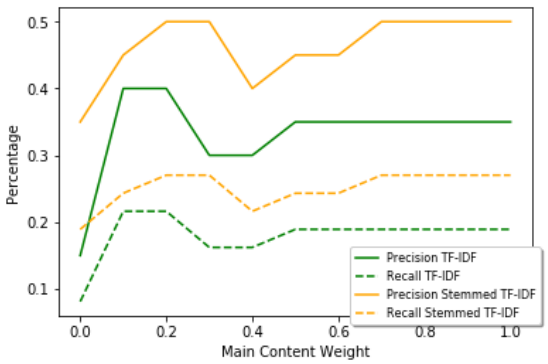

Fig 6. The effect of stemming on 20 documents retrieval.

5 Documents Retrieval

The performance of each method in 5 documents retrieval scenario is shown in Figure 7. From the experimental results, we can see that S1 and S2 having the best and stable results with almost 100% and 30% recall. Meanwhile, our proposed method (S3) having lower results in retrieving relevant documents though showing similar performance as we increase the main content's weight.

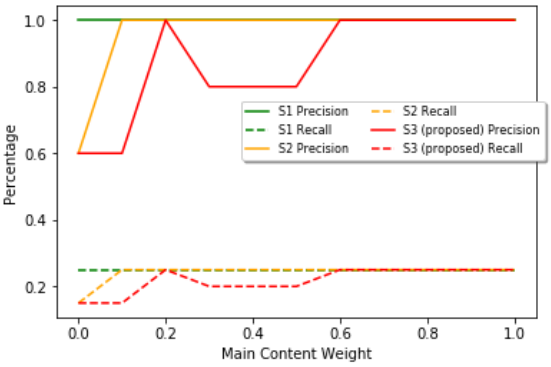

Fig 7. Each method comparison on 5 documents retrieval with simple query.

However, as we inserted more keywords into the query, the results decreased which is shown in Figure 8. From the figure shown, we can see that our proposed method has the highest results on lower weights. But as the weight increased both S3 and S2 have similar results with 40% precision and 5% recall which is 20% lower precision and 25% lower recall from the previous experiment. In this case, we can see clearly that S1 has the lowest result with 20% precision and recall below 5%. From the experiment conducted in 5 documents retrieval, both S2 and S3 outperform S1. Nevertheless, S2 having more stable results.

10 Documents Retrieval

In this case, we increase the amount of the documents retrieval to 10. The performance of each method in 10 documents retrieval scenario is shown in Figure 9. Unlike 5 documents retrieval, in this case, our proposed method has the highest results in explicit queries with 10% higher precision and 3% higher recall followed by S2 and S1.

The keywords addition in the query turns S2 as the best method results with 10% higher precision and 3% higher recall followed by S1 and S3 which have similar performance. In this case, our proposed method shows its best result when the weight is more than 0.7. The results of this case are shown in Figure 10. From these findings, we can conclude that S2 is the best method in 10 documents retrieval with a simple query and our proposed method shows the best results in more complicated queries.

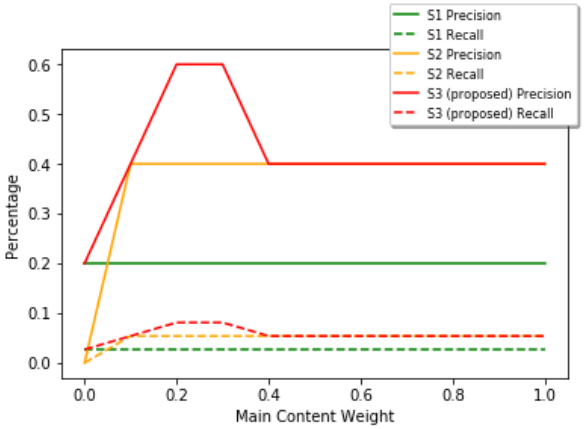

Fig 8. The impact of keywords addition into the query on 5 documents retrieval.

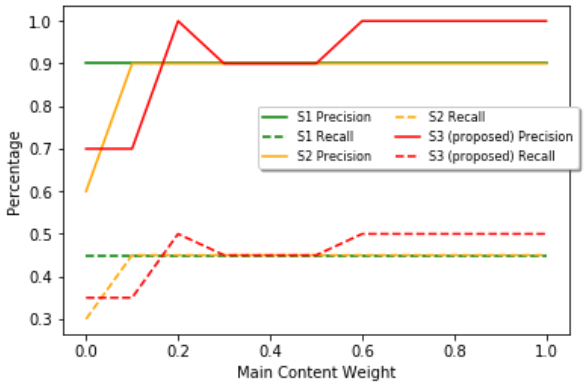

Fig 9. Each method comparison on 10 documents retrieval with simple query.

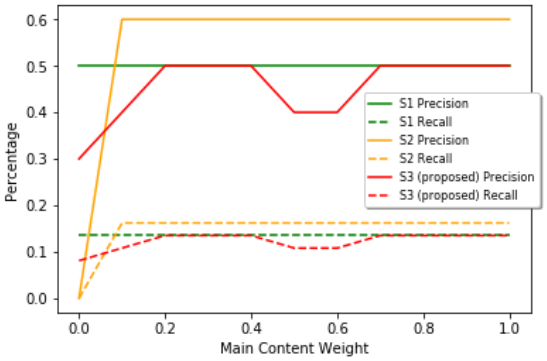

Fig 10. The impact of keywords addition into the query on 10 documents retrieval.

15 Documents Retrieval

15 documents retrieval shows the same pattern as the 10 documents as shown in Figure 11 and Figure 12. In this case, our proposed method have the best result with ~93% precision and ~70% recall in simple queries. Meanwhile, in more complicated queries, the S2 is the best method followed by S3 and S1.

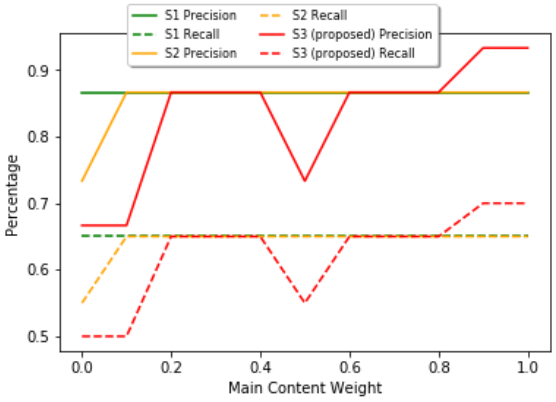

Fig 11. Each method comparison on 15 documents retrieval with simple query.

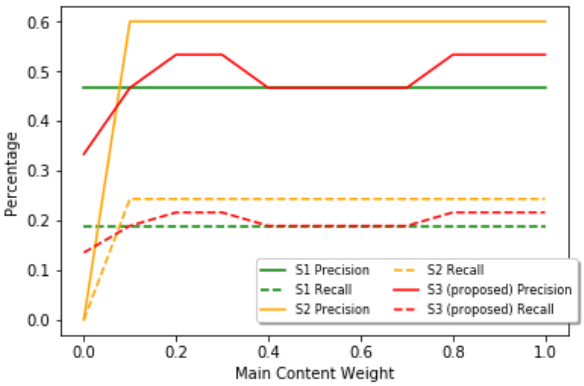

Fig 12. The impact of keywords addition into the query on 15 documents retrieval.

20 Documents Retrieval

Lastly, In this 20 document retrieval with simple queries, we found that all method has the same 70% recall, but S2 outperform S1 and S3 with 20% higher precision. As well as previous experiments, our proposed method show its best performance with 0.7 main content weight. In this case, our proposed method has the same result as S1 generally as shown in Figure 13.

However, from the experiment shown in Figure 14, we found that if more keywords inserted in the queries, S1 has the lowest results. Meanwhile, our proposed method has the highest results with 50% precision and 25% recall, followed by S2.

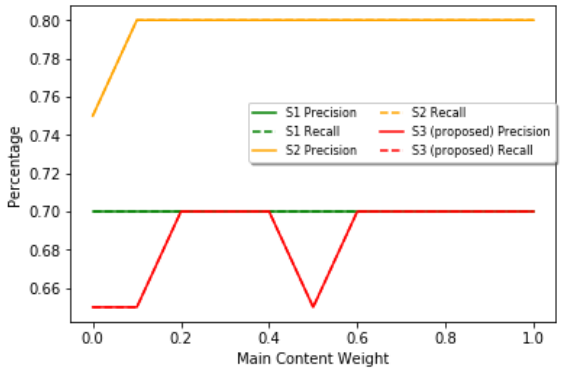

Fig 13. Each method comparison on 20 documents retrieval with simple query.

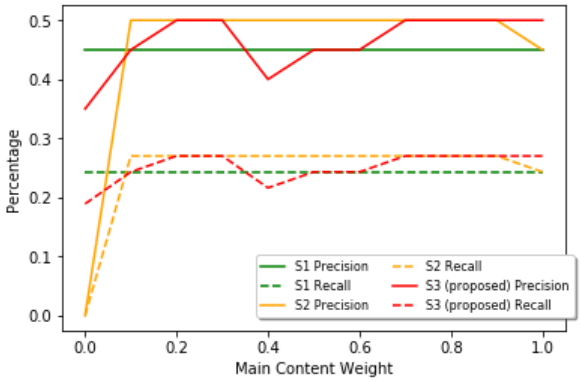

Fig 14. The impact of keywords addition into the query on 20 documents retrieval.

RESULT AND DISCUSSION

The results of TF-IDF query expansion with rocchio pseudo-relevance feedback shows that every method used in this experiment has interesting results with 60-100% precision for all *n* documents and 10-30%, 30-50%, 50-65%, 65%- 70% recall for 5, 10, 15, and 20 documents retrieval respectively. Nevertheless, there is a slight improvement up to 10% precision and 5% recall in S3 for 10 and 15 documents.

However, when more keywords inserted into the query, the results decreased with end results of precision 20%-67% and recall 4-8%, 9-15%, 13-20%, 20%-27% for 5, 10, 15, and 20 documents respectively. These low recall results are caused by TF-IDF nature which can not extract different keywords with similar meaning properly. In this case, both S2 and our proposed method outperform S1 in all documents except for S3 in 10 documents retrieval. S2 shows more stable and better results for all *n* documents and our proposed method having the best result in 5 documents.

From those two query cases, we can point out the best threshold value for S2 and S3. For all *n* documents, any weight of *matan* in S2 shows better and stable results than others. This shows for simple constructed query, *isnad* can be considered as a noise in information retrieval. Meanwhile, our proposed method has more diverse results depends on the complexity of the query. But generally, our proposed method will produce higher results as we increased the main weight content.

Although it is not better than S2 overall performance, our proposed method show slight improvement over the S1 which component is not separated and outperforms S2 in some documents and specific cases. Our S3 unstable results is caused by the low component separation performance which is NER (in this study) with only 72% of accuracy. Also, the chosen component was selected from the small observation conducted in this study. For a better result, we can use a better separation process and components that is more scientifically proven.

TF-IDF Rocchio and Doc2Vec

The experiment conducted on 4590 documents shows that content weighting on TF-IDF pseudo-relevance feedback query expansion outperform Doc2Vec as shown in Figure 15. From the figure we can infer that S1, S2, and S3 scenario doesn't work really well on Doc2Vec.

Altough we implement and execute Doc2Vec as the best practice written on the documentation stated on Gensim library, yet

can not retrieve any relevant documents for all component separation scenarios. Unlike TF-IDF which weighting each word frequency, Doc2Vec represent document as vector space regardless its length. To best of our knowledge, these bad results of Doc2Vec are due to lack of datasets. Therefore the model can not retrieve relevant document because do not know the vocabulary. These results also show that we need more experiment to verify how Doc2Vec affect our proposed method.

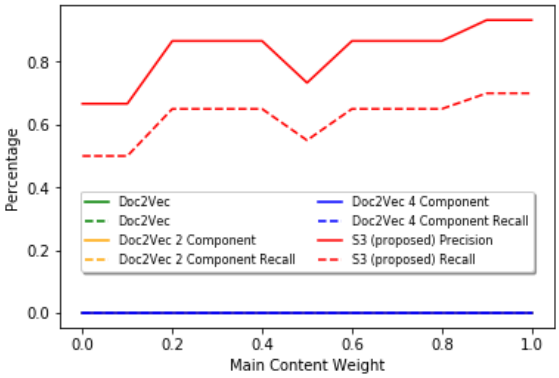

Fig 15. Doc2vec – TF-IDF comparison on 15 documents retrieval.

Existing Method Comparison

Lastly, we compare our method with other hadith retrieval system, particularly which was proposed by Aulia [3]. Aulia proposed a word-based with Nazief stemming procedure in her retrieval system. From the Figure 16 we can see that Aulia's keyword-based retrieval system outperform our proposed method. But, in term of context of the hadith, we can see that our proposed method deliver much better results.

This means that in term of the number of keyword existence in a document, our proposed have a lower result. Nevertheless, our proposed method is superior to hers in searching relevant hadith based on the intent of the query.

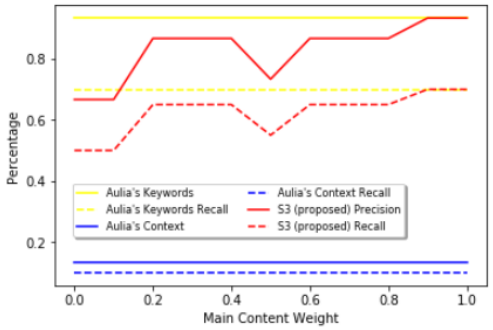

Fig 16. Keyword and context based method Comparison.

CONCLUSION

In this paper, we show hadith component separation and weighting on translated hadith of Abu Daud which contains 4590 documents. We present that hadith component separation affect the retrieval performance as *isnad* can be considered as noise in vector space of a query. Therefore, *isnad* should be removed from vector space since any weight above 0 in S2 will produce better results. Meanwhile, in S3, main content should be weighted between 0.8 – 0.9 to keep other component values in the retrieval process. The experimental results also show that stemmed datasets produce better performance in pesudo-relevance feedback with rocchio algorithm and TF-IDF. Also, we present that our proposed method can retrieve relevant hadiths with similar context with the intent of a query. However, Doc2Vec can not retrieve relevant documents because of the lack of vocabulary and datasets. Thus, combining this with a pre-trained model should have produced a better performance.

Although 2 separated components (S2) has the best overall results generally, 4 separated components (S3) shows better results in some specific cases with up to 100% precision and 70% recall. Also, since our NER component separation method which only has 72% of accuracy have not optimized yet, it indicates that S3 have more room for improvement for better results.

REFERENCES

[1] I. Rasyidi, A. Romadhony, and A. T. Wibowo, "Indonesian Hadith Retrieval System using thesaurus," Proceeding - 2013 Int. Conf. Comput. Control. Informatics Its Appl. "Recent Challenges Comput. Control Informatics", IC3INA 2013, pp. 285–288, 2013.

[2] N. N. Amirah, T. M. Rahim, Z. Mabni, H. M. Hanum, and N. A. Rahman, "A Malay Hadith translated document retrieval using parallel Latent Semantic Indexing (LSI)," 2016 3rd Int. Conf. Inf.

Retr. Knowl. Manag. CAMP 2016 - Conf. Proc., pp. 118–123, 2017.

[3] A. Aulia, D. Khairani, and N. Hakiem, "Development of a retrieval system for Al Hadith in Bahasa (case study: Hadith Bukhari)," 2017 5th Int. Conf. Cyber IT Serv. Manag. CITSM 2017, 2017.

[4] N. A. Rahman, Z. A. Bakar, and T. M. T. Sembok, "Query expansion using thesaurus in improving Malay Hadith retrieval system," Proc. 2010 Int. Symp. Inf. Technol. - Syst. Dev. Appl. Knowl. Soc. ITSim'10, Vol. 3, pp. 1404–1409, 2010.

[5] I. Humaini, T. Yusnitasari, L. Wulandari, D. Ikasari, and H. Dutt, "Informatian Retrieval of Indonesian Translated version of Al Quran and Hadith Bukhori Muslim," 2018 Int. Conf. Sustain. Energy, Electron. Comput. Syst. SEEMS 2018, pp. 1–5, 2019.

[6] A. Zayd, "Hadith: Muhammad's Legacy in the Medieval and Modern World, 2nd ed.," Am. J. Islam. Soc. Sci., Vol. 36, No. 2, pp. 64–73, 2019.

[7] Q. V. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," Proc. 24th Int. Conf. World Wide Web - WWW '15 Companion, Vol. 32, pp. 29–30, 2014, [Online]. Available: https://cs.stanford.edu/~quocle/paragraph_vector.pdf%0Ahttp://dl.acm.org/citation.cfm?doid=2740908.2742760.

[8] J. J. Rocchio, "Relevance feedback in information retrieval," in The {SMART} Retrieval System -- Experiments in Automatic Document Processing, G. Salton, Ed. Englewood Cliffs, NJ: Prentice Hall, 1971, pp. 313–323.

[9] C. Buckley and G. Salton, "Optimization of relevance feedback weights," in Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '95, 1995, pp. 351–357.

[10] A. T. Ni'mah and A. Z. Arifin, "Perbandingan Metode Term Weighting terhadap Hasil Klasifikasi Teks pada Dataset Terjemahan Kitab Hadis," Rekayasa, Vol. 13, No. 2, pp. 172–180, 2020.

[11] N. Rastogi, P. Verma, and P. Kumar, "Evaluation of information retrieval performance metrics using real estate ontology," in Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, 2020, No. Icssit, pp. 102–106.

[12] D. Widiyatmoko and A. Setiyono, "Information retrieval of physical force using the Tf-IdF," in 2019 International Conference on Information and Communications Technology, ICOIACT 2019, 2019, pp. 519–522.

[13] I. Khusna, Arfiani Nur; Agustina, "Implementation of Information Retrieval Using Tf-Idf Weighting Method On Detik.Com's Website," in 2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA), 2019, pp. 1–4.

[14] A. Librian, "Sastrawi," 2016. https://github.com/sastrawi/sastrawi (accessed Jan. 07, 2020).

[15] D. P. Andita Dwiyoga Tahitoe, "Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming," J. Ilm., pp. 1–15, 2010.

[16] H. T. Arifin, Agus Zainal; Mahendra, Putu Adhi Kerta Mahendra; Ciptaningtyas, "ENHANCED CONFIX STRIPPING STEMMER AND ANTS ALGORITHM FOR CLASSIFYING NEWS DOCUMENT IN Representation of Textual," Technology, pp. 149–158, 2007.

[17] A. Jelita, "Effective Techniques for Indonesian Text Retrieval," Ph.D Thesis, pp. 1–286, 2007, [Online]. Available: https://researchbank.rmit.edu.au/view/rmit:6312.

.