

## TERM WEIGHTING BASED ON POSITIVE IMPACT FACTOR QUERY FOR ARABIC FIQH DOCUMENT RANKING

Rizka W. Sholikah, Dhian S. Y. Kartika, Agus Zainal Arifin, and Diana Purwitasari

Informatics Department, Faculty of Information and Technology, Institut Teknologi Sepuluh Nopember, Jl. Raya ITS Kampus Sukolilo, Surabaya, 60111, Indonesia

E-mail: rizka.wakhidatus15@mhs.if.its.ac.id, dhian.satria15@mhs.if.its.ac.id, agusza@cs.its.ac.id, diana@if.its.ac.id

### Abstract

Query becomes one of the most decisive factor on documents searching. A query contains several words, where one of them will become a key term. Key term is a word that has higher information and value than the others in query. It can be used in any kind of text documents, including Arabic Fiqh documents. Using key term in term weighting process could led to an improvement on result's relevancy. In Arabic Fiqh document searching, not using the proper method in term weighting will relieve important value of key term. In this paper, we propose a new term weighting method based on Positive Impact Factor Query (PIFQ) for Arabic Fiqh documents ranking. PIFQ calculated using key term's frequency on each category (mazhab) on Fiqh. The key term that frequently appear on a certain mazhab will get higher score on that mazhab, and vice versa. After PIFQ values are acquired, TF.IDF calculation will be done to each words. Then, PIFQ weight will be combine with the result from TF.IDF so that the new weight values for each words will be produced. Experimental result performed on a number of queries using 143 Arabic Fiqh documents show that the proposed method is better than traditional TF.IDF, with 77.9%, 83.1%, and 80.1% of precision, recall, and F-measure respectively.

**Keywords:** Document Ranking, Arabic, Term Weighting, Query, PIFQ.

### Abstrak

*Query* menjadi salah satu faktor penentu dalam pencarian dokumen. Dalam sebuah *query* terdiri dari beberapa kata, dimana salah satunya menjadi *key term*. *Key term* adalah kata yang memiliki nilai informasi dan bobot lebih tinggi dibandingkan kata lain. Hal tersebut berlaku untuk semua jenis dokumen teks, termasuk dokumen fiqh berbahasa Arab. Penitik beratan pada *key term* dalam proses pembobotan kata memungkinkan terjadinya peningkatan relevansi pencarian. Di dalam pencarian dokumen fiqh berbahasa Arab, jika metode pembobotan kata yang digunakan tidak tepat, *key term* tidak akan memberikan pengaruh berarti. Oleh karena itu diusulkanlah sebuah metode pembobotan baru pada kata berbasis *Positive Impact Factor Query* (PIFQ) untuk perangkaan dokumen fiqh berbahasa arab. PIFQ dihitung menggunakan frekuensi kemunculan *key term* pada setiap kategori (mazhab) dalam fiqh. Semakin tinggi frekuensi *key term* tersebut pada suatu mazhab semakin tinggi pula nilainya pada mazhab tersebut, begitu pula sebaliknya. Setelah didapat nilai PIFQ, kemudian dilakukan perhitungan TF.IDF untuk setiap kata. Selanjutnya bobot PIFQ akan dikom-binasikan dengan TF.IDF sehingga menghasilkan bobot baru untuk masing-masing kata. Hasil dari pengujian yang dilakukan pada sejumlah *query* dengan 143 dokumen fiqh berbahasa Arab menunjukkan bahwa metode usulan dapat lebih unggul jika dibandingkan metode TF.IDF, dengan nilai *precision*, *recall*, dan *F-measure* masing-masing sebesar 77,9%, 83,1%, dan 80,1%.

**Kata Kunci:** Perangkaan Dokumen, Bahasa Arab, Pembobotan Kata, Query, PIFQ.

### 1. Introduction

Documents ranking is one of the research topics in information retrieval. One of its implementation is to sort the query results. The top result is considered as the most relevant according to query entered by user.

Many researches about documents ranking and sorting have been done before, such as N-gram

method, to find relevant documents by matching the query and the document itself [1]. Another research used TF.IDF weighting and represent it to a Vector Space Model (VSM) [2]. TF (Term Frequency) is a method to set the weight of each term by calculating term's frequency in a document. While IDF (Inverse Document Frequency) consider that the fewer term appear in multiple documents, then the higher the weight of

that term. Those two methods combined and called as TF.IDF term weighting [3]. Basically, TF.IDF only counts terms occurrence and give positif discrimination to rare terms in a document; then Fauzi (2014) [4] proposed an improved method called IBF (Inverse Book Frequency) which consider the rare terms in a book. IBF uses the same principal method as IDF, but with difference scope. IBF also proved to have higher precision and recall than basic TF.IDF. Another research about term weighting in Arabic document also done by Khadijah (2015) [5]. The method assume that the relevancy of query and search results also depend on user’s subjectivity, in this case called as preference. Therefore, they proposed IPF $\alpha$  method to accommodate such requirement. This method proved can be applied in documents search and has higher recall than others methods.

In addition to weighting methods, another documents categorizing researches using supervised method has also been developed by Emmanuel (2013) [6]. They proposed Positive Impact Factor (PIF) based on assumption that “positive impact from a certain feature on a certain category could be used to calculate its own negative impact to the other categories”. The result shows that PIF can improve accuracy of documents categorizing while compared with the other existing methods.

In a query that contains several words, there must be a word that have higher information and value than the others. That word is called key term. Using key term in term weighting process could led to an improvement on result’s relevancy. In Arabic Fiqh document searching, term weighting that use improper method will relieve important value from key term.

In this paper, we propose a new term weighting method based on Positive Impact Factor Query (PIFQ) for Arabic Fiqh documents ranking. This method takes note to key term on each query and will give higher weight to it rather than the others. This method is expected to improve documents relevancy compared to existing methods. It may also can be implemented to rank another language documents.

## 2. Methodology

In this study, we used Arabic Fiqh data set from Al-Mahtabah As-Shamela that can be

TABLE 2.  
KEY TERM FREQUENCY

Mazhab	Frequency
Hanafiyah	15
Malikiyah	10
Syafi’iyah	6
Hanabilah	7

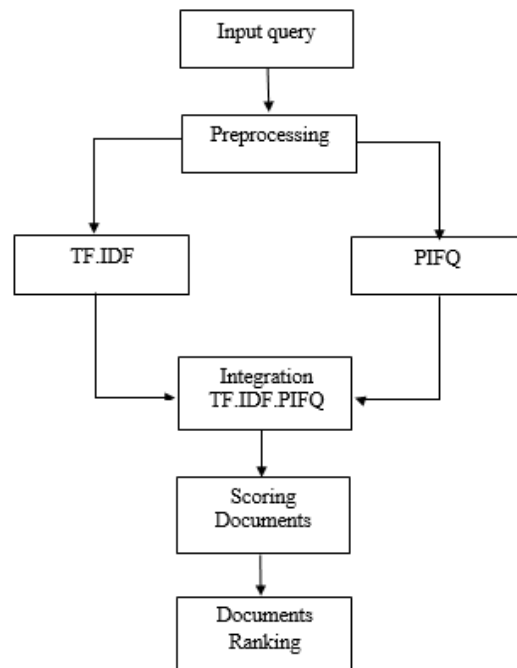


Figure 1. Phases in proposed method

download from <http://shamela.ws/>. The dataset consist of 4 mazhab, Hanafiyah, Malaikiyah, Syafi’iyah, and Hanabilah. Mazhab is kind of methodology that used by Islamic leader to explain the law in Islam so that everyone knows their principal, part, rules, and boundaries [7]. Each mazhab has their own base rule to explain and solve the problems. There are 143 documents that used in this paper, one document is represented by one page. Every documents are took from different books in different mazhab. Figure 1 shows every phases of proposed method include preprocessing.

### Preprocessing

We used 7 different queries in this study. Both queries and documents are trained through the preprocessing phase. The first stage of preprocessing is tokenizing. Tokenizing was done to eliminate space, punctuation, and numbers so that the document will consist by a set of single word. In this phase, the vowels in every word are removed. The next step is to do stop-word removal by removing the words (terms) that has no valuable information. The terms that constantly appear in

TABLE 2.  
KEY TERM PIFQ

Mazhab	PIFQ
Hanafiyah	1,218
Malikiyah	1,133
Syafi’iyah	1,075
Hanabilah	1,088

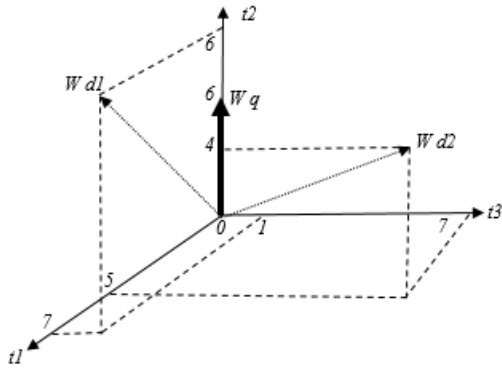


Figure 3. The illustration of VSM

every documents with high frequency can be categorizing as stop-word. In this paper, we used Arabic stop-word list that contain of 13.016 words. The stop-word list can be download from <http://arabicstemmer.codeplex.com/>.

After we got the valuable terms, then each term will going through normalization process. Normalization in Arabic text is important to be done because Arabic text has various way to write the same word. Normalization can be done by following this step [8, 9]: 1) convert term ,(!) ,(!) ,(ع) ,(س) ,(ة) into alif (!); 2) convert ta marbutoh (ة) into ha (ه); 3) convert ya (ي) into ya (ي).

The last stage is stemming, that used to obtain the root in each words. In this study we used Light Stemmer [10]. Light stemmer is one of the method to find root in Arabic without using dictionary. This method get the root only by removing conjunction for example wa (و), some prefixes (ف, ك, ل, ب, ل, ال) and suffixes (ها, ون, وا, ين, ان, به, ا, ة) [9]. To perform stemming and normalization in Arabic, we used library from Apache Lucene in Java that can be download from <http://lucene.apache.org/>.

### TF.IDF

The common method for term weighting is TF.IDF. TF (Term Frequency) is one of the method to get weight of terms by calculating the frequency of terms in a document [3, 11]. IDF (Inverse Document Frequency) assume that each term which rarely appear on the multiple document in data set has higher value [3, 11]. TF for every terms  $t_i$  in documents  $d_j$  can be calculated using equation(1).

$$W_{TF}(t_i, d_j) = f(t_i, d_j) \quad (1)$$

In a corpus consisting of  $D$  documents, there are  $d_{(t_i)}$  documents that contained terms  $t_i$ . The IDF can be calculated as defined in equation(2).

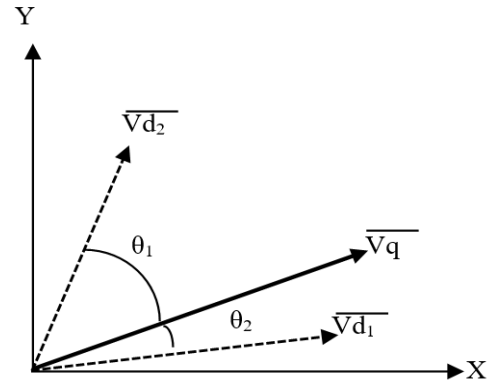


Figure 2. Cosine similarity illustration

$$W_{IDF}(t_i) = 1 + \log\left(\frac{D}{d_{(t_i)}}\right) \quad (2)$$

TF.IDF weight calculation is done by multiplying the equation 1 with 2 resulting in equation(3).

$$W_{TF.IDF}(t_i, d_j) = f(t_i, d_j) \times \left(1 + \log\left(\frac{D}{d_{(t_i)}}\right)\right) \quad (3)$$

### Proposed Method

PIFQ (Positive Impact Factor Query) is a modification of the PIF (Positive Impact Factor) method used to perform documents categorizing [6]. PIF using the occurrence of each terms in every category to get the weight for creating a classifier model. PIF method can be calculated by using equation(4).

$$W_{PIF} = \log\left(\frac{F_{M_i}}{\sum_{j=1}^n F_{M_j}} + 1\right), j \neq i \quad (4)$$

Based on PIF, this proposed method using the occurrence of key term in every category (mazhab) to improve the relevancy between query and search result. Key term itself is a word in a query that has higher value and information than the other words in that query. In this study, the first word in the query is considered as a key term. The idea of PIFQ is to increase the similarity between query vector and documents vector by increasing the weight of the documents using key term. PIFQ provides higher value in document that located in high key term frequency mazhab. It is based on assumption that those documents has higher probability as relevant documents than documents in other mazhab.

To calculate PIFQ, it will be seen whether the terms  $t_i$  is key term  $k$  or not. If  $t_i$  is key term  $k$  then the PIFQ will be calculated. If not, the value will be assign by 1. In a corpus of Arabic Fiqh documents which consists of four mazhab  $M = \{M_1, M_2, M_3, M_4\}$ , frequency of the key term that

TABLE 3.  
CONFUSION METRICS

	Relevant	Not relevant
Retrieve	TP	FP
Unretrieve	TN	FN

occur in every mazhab  $F_{Mi}$  will be calculated. Then the weight of each terms in every documents  $d$  according to their mazhab  $M_i$  will also be calculated. PIFQ weight calculation can be seen in equation(5).

$$W_{PIFQ}(t_i) = \begin{cases} 1 + \log\left(\frac{F_{Mi}}{\sum_{j=1}^4 F_{Mj}} + 1\right), j \neq i, t_i = k \\ 1, t_i \neq k \end{cases} \quad (5)$$

For the query الجمعة صلاة في المسجد, since the first word is الجمعة, then that term is considered as the key term of the query. After we get the key term, the next step is to calculate the frequency of key term in every mazhab. Table 1 shows an example of the calculation of frequency.

After that, PIFQ value in every mazhab can be calculated using equation(5). The example of calculation if key term is located in mazhab Hanafiyah can be seen below.

$$W_{PIFQ}(\text{الجمعة}) = 1 + \log\left(\frac{15}{10 + 6 + 7} + 1\right) = 1,218$$

Table 2 shows the PIFQ in every mazhab. The key term value in every documents can be different depend on where the document is located. While, for the non-key term the value will be assign by 1.

The final step is combining TF.IDF with PIFQ by multiplying the weight of each word from TF.IDF and PIFQ calculation. The formula for determining the weight using the proposed method can be seen in equation(6).

From equation(6) we can see that if term  $t_i$  equals key term  $k$ , the term will get a higher weight than other terms. Thus, if the documents contain more key terms, the weight will be superior to other documents. By using the proposed method, the value of key term in the documents that belong to different madzhab will be different. Since this

TABLE 4.  
LIST OF QUERY

ID Query	Query
Q1	الماء الذي ينجس والذي لا ينجس
Q2	المسخن الماء
Q3	الوضوء من النوم
Q4	الجنابة غسل
Q5	عورة الحرة
Q6	الصدقة دراهم
Q7	الزكاة ليست الأرز

method use the density of key term in every madzhab by calculating the frequency, then if the document located at madzhab that have high frequency of key term, the weighting of key term will also be high, and vice versa.

$$W(t_i) = \begin{cases} W_{TF.IDF}(t_i, d_j) \times \left(1 + \log\left(\frac{F_{Mi}}{\sum_{j=1}^4 F_{Mj}} + 1\right)\right), j \neq i, t_i = k \\ W_{TF.IDF}(t_i, d_j) \times 1, t_i \neq k \end{cases} \quad (6)$$

### Vector Space Model (VSM)

After getting the weight for each terms, then the document will be represented in a vector space using VSM. VSM is a model used to measure the similarity between document and a query. Query and documents considered as vectors in n-dimensional space, where  $k$  is the sum of all terms in the lexicon. Lexicon is a list of all the terms in the index. After that, the cosine angle of the two vectors, namely  $Wd$  of each document and  $Wq$  of the query, will be calculated. VSM is usually used when the terms weighting is done with TF.IDF method. This is because TF.IDF method allows the similarity weight on the document. Every term  $t$  in a document described as one dimension, so that if there are three terms would form a 3 dimension. Figure 2 shows an illustration of the VSM.

### Similarity Measurement

One of the method to measure the similarity between documents is cosine similarity [12]. This measure calculates the cosine of the angle between

TABLE 5.  
EXPERIMENT RESULT

Query	IDF			TF.IDF			IDF.PIFQ			TFIDFPIFQ		
	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
Q1	43.8	50.0	46.7	56.3	68.8	61.9	43.8	50.0	46.7	<b>75.0</b>	<b>75.0</b>	<b>75.0</b>
Q2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Q3	<b>66.7</b>	<b>83.3</b>	<b>74.1</b>	58.3	75.0	65.6	<b>66.7</b>	75.0	70.6	50.0	66.7	57.1
Q4	50.0	<b>100.0</b>	66.7	60.0	90.0	72.0	60.0	<b>100.0</b>	75.0	<b>70.0</b>	90.0	<b>78.8</b>
Q5	<b>100.0</b>	100.0	100.0	85.7	100.0	92.3	<b>100.0</b>	100.0	100.0	<b>100.0</b>	100.0	100.0
Q6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Q7	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
Average	72.9	83.3	76.8	72.9	<b>83.4</b>	77.4	74.3	82.1	77.5	<b>77.9</b>	83.1	<b>80.1</b>

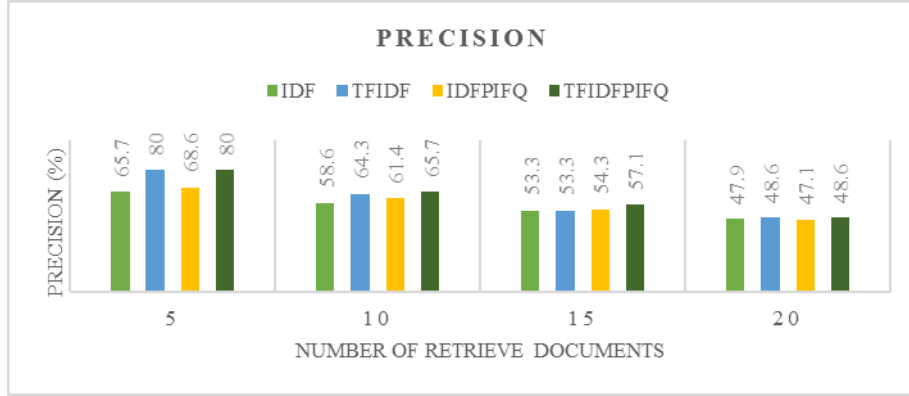


Figure 4. Precision for variation of displayed documents

the two vectors. The use of cosine similarity in text matching has angle limitation between  $0^\circ$  and  $90^\circ$ . This is because in text matching the similarity of documents cannot be negative.

Suppose there are two document vectors  $d_1$ ,  $d_2$  and query vector  $q$ . Cosine similarity will calculate the value of  $\theta$  of each document to the query  $q$ . For each word in the document which has a weight  $W(t_i, d_j)$  and each word in a query that has a weight  $W(t_i, q)$  calculation cosine similarity can be done by applying the formula in equation(7), while for an illustration of cosine similarity can be seen in Figure 3.

$$\begin{aligned} \cos(\theta) &= \frac{d \cdot q}{\|d\| \times \|q\|} \\ &= \frac{\sum_{i=1}^N W(t_i, q) \cdot W(t_i, d_j)}{\sqrt{\sum_{i=1}^N |Wq|^2} \cdot \sqrt{\sum_{i=1}^N |Wd|^2}} \end{aligned} \quad (7)$$

The result of the similarity between documents and query by using cosine similarity will yield a value between 0 and 1. 0 indicates the documents and the query has absolutely nothing in common and 1 indicates that the document and the query is identical.

### 3. Result and Analysis

The proposed method is evaluated by using precision, recall, and F-measure. Precision, recall, and F-measure are commonly used to evaluate performance in Information Retrieval (IR). The experiment in other methods [4, 5, 6] also using precision, recall, and F-measure to evaluate the method's performance. Based on Table 3 precision, recall, and F-measure are defined as Equation 7, 8, and 9 each.

Precision can be described as the number of documents relevant ( $tp$ ) from the total of document that have been retrieve ( $tp + fp$ ). Precision is used to measure the effectiveness of IR systems

(equation 8). Recall is used to measure relevancy of the system (equation 9). This approach can be calculated as the number of doc-ument relevant and retrieve ( $tp$ ) from total number of relevant documents in collections ( $tp + tn$ ). The F-measured approach can be created by combining precision and recall as shown in equation(10).

$$precision = \frac{tp}{tp + fp} \quad (8)$$

$$recall = \frac{tp}{tp + tn} \quad (9)$$

$$Fmeasure = \frac{2 \times recall \times precision}{precision + recall} \quad (10)$$

We used 7 queries to test our proposed method. The queries can be seen in Table 4. Precision, recall, and F-measure will be performed for each query.

In this experiment we used TF.IDF.PIFQ as term weighting. Then we will compare the result of proposed method with IDF, IDF.PIFQ, and TF.IDF, so that the relevancy of proposed method can be observed. Result of the experiment can be seen in Table 5. From the table we know that our proposed method has higher precision values than IDF and IDF.PIFQ in Q1 with 75.0% and Q4 with 70.0%. While compared with TF.IDF, the proposed method has higher precision for Q1, Q4, and Q5, with 75.0%, 70.0%, and 100.0% respectively. Only in Q3 the precision value of proposed method is lower than the three other methods.

The recall been calculated by retrieving 20 documents. The result using IDF, IDF.PIFQ, TF.IDF, and TF.IDF.PIFQ produce the same value at 4 query, i.e. Q2, Q5, Q6 and Q7. In Q1 recall of the proposed method has higher value than the three other methods, with 75.0%. Whereas, in Q3

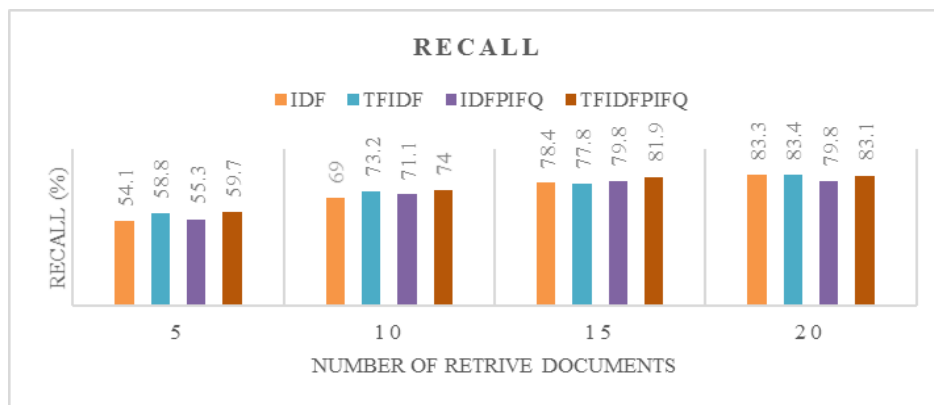


Figure 5. Recall for variation of displayed documents

recall value using IDF method is the highest, with 83.3%. The similarity result in some query caused by the four methods retrieve the same number of relevance documents, but has the different order. The TF.IDF.PIFQ method placed the documents that has high frequency of key term above the lower ones.

The average results showed that the proposed method, TF.IDF.PIFQ has higher value in precision and F-measure, with 77.9% and 80.1%. But the average of recall value is smaller than TF.IDF, the proposed method with 83.1% and TF.IDF with 83.4%.

The experiment also conducted by calculate precision and recall with variation of the number of documents retrieved to user. The first experiment displayed 5 documents relevant, then 10 documents, 15 documents, and 20 documents. For each variety will be calculated the precision and recall for IDF, IDF.PIFQ, TF.IDF and TF.IDF.PIFQ method. The experiment result can be seen in Figure 4 and 5. From Figure 4 we know that the more document that displayed to user, the precision value produced by all methods will decrease. When system retrieve 10 and 15 documents that displayed to user, TF.IDF.PIFQ method got higher precision than other methods, with 65.7% and 57.1% respectively. Whereas, when retrieve 5 and 20 documents the proposed method has the same value with TF.IDF, 80% and 48.6% each.

The result of recall can be seen in Figure 5. For all methods the more documents that displayed to user the result value will increase. The result indicates that our proposed method has higher recall than the others method, with 59.7% for 5 documents, 74% for 10 documents and 81.9% for 15 documents. Only in variety number of documents 20, TF.IDF.PIFQ has 83.1% lower than TF.IDF with 83.4% and IDF 83.3%.

Figure 4 and 5 also shows that adding the PIFQ in IDF method can give positive result in

precision and recall. The precision of IDF.PIFQ when retrieve 5, 10 and 15 documents has higher value than IDF method. The recall also show that in 5, 10, and 15 documents using IDF.PIFQ has higher recall than only using IDF. Therefore, we can conclude that adding PIFQ can increase the result of IDF method by considering the density of key term in every mazhab.

In TF.IDF.PIFQ, high key term frequency mazhab has higher probability contain relevant documents than other mazhab. That condition affects the search result, and can increase the precision and recall. But if there are documents relevant that located in low key term frequency mazhab, the order will be lower than non-relevant documents that contain key term in high key term frequency mazhab. This factor can decrease the precision and recall of the method.

In some queries Q2 and Q6, the query's terms has high occurrence in relevant documents, but does not or just rarely occurre in other documents. So in this condition, implementing all method will led to the same result.

Overall, based on the experiment result by comparing method using precision, recall, and F-measure, we can conclude that TF.IDF.PIFQ method is superior when compared to the method IDF, IDF.PIFQ and TF.IDF. It also prove that the proposed method can be used to improve the relevance of searching result in Arabic Fiqh Documents. However, this method has weakness when the distribution of key terms in each mazhab are equals. So that PIFQ value for key term in every mazhab will be the same. This will reduce the influence of the importance of key terms in the search process.

#### 4. Conclusion

This paper shows a new terms weighting method which observe the impact of key term density on each mazhab. This new method

combines PIFQ and TF.IDF to calculate the weight of each terms. This new method proved to be able to be implemented on terms weighting on Arabic Fiqh document ranking. It can be seen on the experimental result which shows that TF.IDF.PIFQ method has better precision and F-measure than IDF, IDF.PIFQ and TF.IDF, which are 77.9% and 80.1% respectively. Moreover, using various number of documents displayed to the user, TF.IDF.PIFQ is able to get higher precision and recall value compared to the other three methods by the variation of 10 and 15 documents. The experiment also shows that adding PIFQ can increase precision and recall of IDF method. The average result of precision and F-measure using IDF.PIFQ is higher than using IDF, with 74.3% and 77.5% respectively. Besides used on Arabic documents, this method could also be used on documents with other languages.

## References

- [1] S. H. Mustafa and Q. A. Al-Radaideh, "Using N-Grams for Arabic Text Searching," *Journal of The American Society for Information Science and Technology*, pp. 1002-1007, 2004.
- [2] F. Harrag, A. Hamdi-Cherif and E. El-Qawasmeh, "Vector Space Model for Arabic Information Retrieval - Application to Hadith Indexing," *Proceedings of the First IEEE Conference on the Application of Digital Information and Web Technologies*, pp. 107-112, 2008.
- [3] G. Salton and B. Christopher, "Term-weighting approach in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513-523, 1988.
- [4] M. A. Fauzi, A. Z. Arifin and A. Yuniarti, "Term Weighting Berbasis Indeks Buku dan Kelas untuk Perangkingan Dokumen Berbahasa Arab," *Lontar Komputer*, vol. 5, no. 2, 2014.
- [5] K. F. H. Holle, A. Z. Arifin and D. Purwitasari, "Preference based term weighting for arabic fiqh document ranking," *Journal of Computer Science and Information*, vol. 8, no. 1, 2015.
- [6] E. M, S. M. Khatri and R. B. D.R., "A Novel Scheme for Term Weighting in Text Categorization : Positive Impact Factor," *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2292-2297, 2013.
- [7] "Mazhab," Wikimedia, [Online]. Available: <http://id.wikipedia.org/wiki/Mazhab>. [Retrieved December 25, 2015].
- [8] A. Odeh, A. Abu-Errub, Q. Shambour and N. Turab, "Arabic Text Categorization Algorithm using Vector Evaluation Method," *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 6, no. 6, 2014.
- [9] H. K. H. Chantar, "New Techniques for Arabic Documents Classification", Ph.D Thesis, School of Mathematical and Computer Science, Heriot-Watt University, 2013.
- [10] L. S. Larkey, L. Ballesteros and M. E. Connell, "Light Stemming for Arabic Information Retrieval," *Arabic computational morphology Springer*, pp. 221-243, 2007.
- [11] G. Salton, Automatic text processing : The Transformation, analysis, and Retrieval of Information by Computer, Addison-Wesley, 1989.
- [12] S. Tata and J. M. Patel, "Estimating the selectivity of tf-idf based cosine similarity predicates," *SIGMOD Record*, vol. 36, no. 4, pp. 75-80, 2007.