International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Using TF-IDF to Enhance Information Retrieval in Hadith Corpus

Dr. Samah Mohamed Osman Hassan[1], Dr. Eric Atwell[2]

[1]Arab Open University, Saudi Arabia, Riyadh
Email: *samahaltoum2004[at]yahoo.com*

[2]University of Leeds, Leeds, UK
Email: *e.s.atwell[at]leeds.ac.uk*

Abstract: ***This paper aims to address the challenge of Information Retrieval from the Hadith, focusing on multiple languages and utilizing a Hadith parallel corpus. The proposed approach involves employing a matching algorithm for the retrieval process. It calculates the weight of words in the query based on their importance and compares them with existing documents that have undergone processing to determine the significance of words in each document. Subsequently, a similarity coefficient is computed between the specific query and the existing documents. To enhance performance, the system utilizes a dictionary of words, implementing an inverted index to identify all files containing those words. The proposed solution is designed and evaluated by selecting important concepts, for which manual results have been predetermined independently from the system. The evaluation process measures both average precision and average recall for each language.***

Keywords: Information Retrieval, Hadith, parallel corpus, matching algorithm, similarity coefficient

1. Introduction

Information Retrieval (IR) systems retrieve relevant information relating to a specific query by the user, and this requires the extraction of related unstructured information from data which may be texts, sound, images. In this context, an important problem facing information retrieval, in particular from text files, is reliance on exact matching of the word or words in the query and the same words in a specific text file. This leads in many cases to the loss of results where files contain synonyms with words in the query which may be useful to the user. This dilemma appears in most Information retrieval systems for unstructured text data, and with most languages, especially with regard to the Arabic language. This research will deal with the problem of Information Retrieval from the Hadith across many languages, by building a parallel corpus with multiple languages containing the Hadith in Arabic as well as translated texts in English, French and Russian.

2. Methodolgy

In the following sections we will explain in full details how the work had been done using Hadith corpus [4].We will now examine the structure and implementation of TF-IDF for a set of documents. We will first introduce the mathematical background of the algorithm and examine its behavior relative to each variable. We then present the algorithm as we implemented it.

2.1 Mathematical Framework

Retrieval methodologies assign a measure of similarity between a query and a document. These methodologies are based on the common notion that the more often terms are found in both the document and the query, the more "relevant" the document is assumed to be to the query. Additionally,a retrieval methodologies is an algorithm that takes a query Q and a set of documents D1,D2 , ••• , Dn and identifies the cosine similarity cosine(θ) for each of the documents $1 \leq i \leq n$.Our model is based on the idea of each document can be represent into vector and by the same way represent the query into vector then select method to measure the closeness of any two documents. Therefore we will convert the entire corpus documents into vectors consider that each document(Hadith) represent one vector(v) in the space ,and to avoid the different length of each document we will transform all of them into same length of vectors using the tfidf representation in the sklearn [4][5].so that will be accept the query from the user converted to the vector by the same size of the vector documents then apply the cosine similarity to find the most relevant documents to the user query. For in Figure 3 show that there are 3 documents (d1,d2,d3) and Q in the vector space model each document represent as vector and we notice that d2 and Q are similar because the vector between the two vectors are 0 (Zero) or in another word the cosine(0)=1 and that means there are 100% similarity.

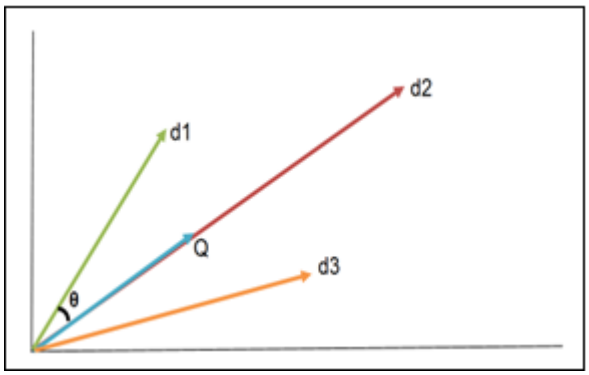

Figure 3: Show the similarity between Q and d2 in the vector space

For the formal definition of and declare the use of weights based on the collection frequency.Weight is computed using the TF_IDF and for that to construct a vector that represent

each document [1],we will use the following formula from (Grossman et al,2012).
tfij=number of occurrences of the term tj in document Di

dfj=number of document which contain tj

idfj=log(N/dfj) where N is the total number of documents

Calculation of the weighting factor(tfidf) for a term in a document is defined as a combination of term frequency(tf),and inverse document frequency(idf).To compute the value of the jth entry in the vector corresponding to document i, The following equation is used:

$$d_{ij} = tf_{ij} \times idf_j \quad (1)$$

we represent our corpus as a group of documents (D1, D2,…..,Dn) When a document retrieval system is used to query a collection of documents with t terms, the system computes a vector D (di1, di2, .. . , din) of size n for each document. The vectors are filled with term weights as described above. Similarly, a vector Q (Wql, W q2,….,Wqn) is constructed for the terms found in the query.

3. Cosine similarity

The vector is a representation of the mathematical deal with the numbers only so when we want to convert the texts to the vectors must use numbers in this study we decided to use a statistical measure tfidf in order to be able to represent each document in the figure of the vectors so that the application can find the angle between two vectors by calculation of the cosine measure of the angle between the two vectors will lead to the similarity value [3]. However ,the cosine value is between [1,0] if the cosine is 0 that means the two vector are orthogonally and there is no similarity between them ,on the other hand If the cosine is 1 that means the two vector are similar or in the same direction to each other and the angle between them are 0 ,in general the cosine value near 1 means small angle and more similar and the value closer to 0 means large angle with less probability of similarity.A cosine similarity (CS) between a query Q and a document Di is defined by the product of the two vectors. Since a query vector is similar in length(n:vector size) to a document vector, this same measure is often used to compute the similarity between two documents:

$$\text{Cosine}(D_i, Q) = \frac{\sum_{j=1}^{n} d_{ij} \cdot q_j}{\sqrt{\sum_{j=1}^{n} d_{ij}^2 \cdot \sum_{j=1}^{n} q_j^2}} \quad (2)$$

To speed up our scanning process instead of scanning the entire collections of documents we created inverted Indexes for the distinct terms mapping each term to the specific posting list which contain the document numbers for all the documents contains that term [9].In order to build the inverted indexes for our collection we use the python dictionary which is consist of two important things keys and values and for our purpose the terms represent the dictionary "keys" which is unique that mean each term represent one "key" ,on the other hand the values can be a list of number which in our case represent the documents numbers Table 1 show the sample from the inverted index of English terms.

Table 1: Display the Inverted Indexes for the English corpus

Index (Key)	Posting List (Value)
Stress	1167
Buyer	[532, 535, 539, 769, 1315]
Pain	[459, 769, 1278]
Struck	[319, 458, 459, 750]
Sinner	[392, 437, 610, 611, 699, 868, 1011, 1126]

4. Feeding algorithm

Convert the collection of the Ahadith from the comma separated files for all the languages generated list of documents to do that a proposed feeding algorithm is used for generate the multi corpus text and feed the corpus into TfidfVectorizer .The python programming is used to implement these algorithm. After run the code only 2030 documents had been generated. Therefore the TfidfVectorizer will compute the TF_IDF weight for each token in the entire corpus. Consequently a sparse matrix of 2030x5313 of type '<class 'numpy.float64'>' with 60379 stored elements in Compressed Sparse Row format. So that mean will end up with only 7,196 distinct terms from the English Hadith text out of 92,583,7 terms and will do the same for all other languages (Arabic,Frecnh and Russin).

In order to be able to store such a matrix in memory but also to speed up algebraic operations matrix / vector, implementations will typically use a sparse representation such as the implementations available in the (scipy.sparse) package. SciPy (pronounced "Sigh Pie") is a collection of mathematical algorithms and functions built work with Python language.

5. Experiment

Search based on a user query(sometimes called ad hoc search because the range of possible queries is huge and not prespecified) is not the only text-based task that is studied in information retrieval [3].A propsed search Algorithm has been design see Figure 3.33 to generate the search engine using the benefit of the TF-IDF ,coefficine similarity and datamining teqniges like stop words ,tokenization and stemming in python and .To implement our algorithm we use Visual studio 2012, Flask, Python and nltk

Search by keyword will return the specific word in our case the specific Ahadith related to the specific concept. but when we search using the Arabic word we get problem of getting zero results when we know we have data like when we search for the concept "الأيمان" will get no result and that because the concept word in the database is written "الايمان" with no (tashkel"تشكيل") so the word will not match, and to solve this problem we generate a python function to take the Arabic word replace each ("أ","!","إ") by the character ("ا") and each("ه")by("ة") see Figure 1.using the same way try for the other languages respectlly.

Figure 1: Snapshot for Arabic search concept "الايمان"

6. Steeming process

After reading the Arabic text before feeding to the TFIDFvectorize consider the stemming process for the entire Arabic text in the paper we use snowballstemmer stemmer from nltk because it contain Arabic stemmer use with python program. After data stemming only 6,882 terms are left and can be considered as distinct terms. Table 2 shows the comparison between the number of terms before and after data stemming. Figure 2 illustrate the change in the number of terms. We do the same for other languages English ,frenc and Russion respectlly.

Table 2: Number of the terms for Arabic text before and after stemming process

Arabic Text	Number of distinct terms
Before	12,853
After	6,882
Percentage in Reduction	46.46 %

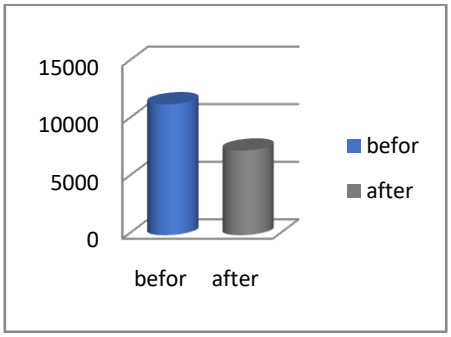

Figure 2: Chart showing the change of number of terms for Arabic before and after stemming process

7. Quantitative Evaluation

There is always an urgent need to improve the performance of Information retrieval system and seek to make it more efficient and effective, and to assess this improvements paperer uses the measurement recall and precision as an indicator of improved performance if the rank rate high for both recall (Eq(1)) and precision ((Eq(2)) mean a clear indication of the efficiency and effectiveness of the system"The best know IR measures are precision and recall " [11] ,so that what will we use to assess our IR system for Hadith [10]. Besides, evaluation is very critical and tedious task in information retrieval system [2].That is why to evaluate our search engine we decide to select a set contain 10 concepts from our corpus called as our gold standard to evaluate and calculate the precision and recall for our system for each languages started with the English, Arabic, French and Russian. In Table 3 we declare our gold standard for the four languages [6].

Table 3: The gold standard for our Search

Russian	French	English	Arabic
переселится	expatriation	emigration	الهجرة
благословит	intention	Intention	النيات
дела	Actes	Deed	الاعمال
откровение	révélation	Revelation	الوحي
мирского	monde	Worldly	الدنيا
он хотел жен	epouser	Marry	ينكحها
веры	foi	Faith	الايمان
наука	Connaissance	Knowledge	العلم
лицемерие	hypocrisie	Hypocrisy	النفاق
омовения	Ablutions	Ablution	الوضوء

The formulas for the precision and the recall values were obtained from [7].The F-measure formula obtained from [8].

$$Recall = \frac{Number\ of\ answer\ given\ by\ the\ system}{\textbf{total number of the correct answers exist in the text}} \quad (3)$$

$$Precision\ (P) = \frac{\textbf{Number of the correct answer given by the system}}{\textbf{Total number of answers given by the system}} \quad (4)$$

In Table 4 show the the differences in Recall, Precision and F-Measure between the languages

$$F - Mesure = 2\frac{R * P}{R + P} \quad (5)$$

Table 4: Display the differences in Recall, Precision and F-Measure between the languages that clearly indicate improvement in search using the cosine similarity match algorithm.

	Arabic %	English %	French %	Russian %
Precision	96.5	98.4	97.5	98
Recall	82	90	91.7	91
F-Measure	88.8	94.2	95	94

8. Conclusion

The result show that while we are searching in the Hadith using the MHC implementing the TF-IDF to determine the most important terms for each document(Hadith) , calculate the coefficient similarity between the query and all the documents to find the most relevant documents to the query on these aspect before apply stemming the result of precision and recall for the Arabic are 75 % and 59 % ,result of precision and recall for English are 89.4 % and 52 % , result of precision and recall for French are 75.5 % and 79.4 % and for Russian precision and recall are 78.5 % and 53.4 respectively. Beside that we notice that there is more improvement when we apply the stemming process which all the result show improvement regards all the languages

starting with precision and recall for Arabic 96.5 % and 82 %, for English are 98.4 % and 90 % ,for French are 97.5 and 91.7 and for Russian are 98 % and 91 % respectively.

References

[1] Zhou, Hai. "Research of text classification based on TF-IDF and CNN-LSTM." Journal of Physics: Conference Series. Vol. 2171. No. 1. IOP Publishing, 2022.
[2] Zuva, K. and Zuva, T., 2012. Evaluation of information retrieval systems. International Journal of Computer Science & Information Technology (IJCSIT), 4(3).
[3] Yang, C.C., 2010. Search engines information retrieval in practice.
[4] Hassan, Samah Mohamed Osman, and E. S. Atwell. "Design requirements for multilingual hadith corpus." International Journal of Science and Research (IJSR) 5.4 (2016):
[5] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Paper, 12(Oct), pp.2825-2830.)
[6] Manning, C.D., Raghavan, P. and Schütze, H., 2008. Introduction to information retrieval (Vol. 1, No. 1, p. 496). Cambridge: Cambridge university press.
[7] Martin, J.H. and Jurafsky, D., 2000. Speech and language processing.International Edition.
[8] Christopher, D.M., Prabhakar, R. and Hinrich, S.C.H.Ü.T.Z.E., 2008. Introduction to information retrieval. An Introduction To Information Retrieval, 151, p.177.
[9] Goker, A. and Davies, J. eds., 2009. Information retrieval: Searching in the 21st century. John Wiley & Sons.
[10] Harrag, F., El-Qawasmah, E. and Al-Salman, A.M.S., 2011, April. Stemming as a feature reduction technique for Arabic text categorization. In Programming and Systems (ISPS), 2011 10th International Symposium on (pp. 128-133). IEEE.
[11] Bar-Ilan, J., 2002. Criteria for evaluating information retrieval systems in highly dynamic environments. In Proceedings of the 2nd International Workshop on Web Dynamics, Honolulu, Hawaii, USA.