ISSN: 0067-2904

Applying Similarity Measures to Improve Query Expansion

Wajih A. Ghani A. Hussain
Department of Computer, College of Science, University of Baghdad, Baghdad, Iraq.

Received: 8/7/2020 Accepted: 27/8/2020

Abstract

The huge evolving in the information technologies, especially in the few last decades, has produced an increase in the volume of data on the World Wide Web, which is still growing significantly. Retrieving the relevant information on the Internet or any data source with a query created by a few words has become a big challenge. To override this, query expansion (QE) has an important function in improving the information retrieval (IR), where the original query of user is recreated to a new query by appending new related terms with the same importance. One of the problems of query expansion is the choosing of suitable terms. This problem leads to another challenge of how to retrieve the important documents with high precision, high recall, and high F measure. In this paper, we solve this problem through applying different similarity measures with the use of English WordNet. The obtained results proved that, with a suitable selection method, we are able to take advantage of English WordNet to improve the retrieval efficiency. The work proposed in this paper is extracting the terms from all the documents and query, then applying the following steps: preprocessing, expanding the query based on English WordNet, selecting the best terms, weighting of term, and finally using the cosine similarity and Jaccard similarity to obtain the relevant documents.

Our practical results were applied on the DUC2002 dataset that contains 559 documents distributed over several categories. The average precision of cosine (for random queries) = 100% whereas the average precision of Jaccard = 84.4 %, and the average recall of cosine = 86.8% whereas the average recall of Jaccard = 73.4%. The average f-measure of cosine = 92%, whereas the average f-measure of Jaccard = 76%.

Keywords: Information Retrieval, Query expansion, Data source search, Cosine Similarity, Jaccard Similarity.

تطبيق مقاييس التشابه لتحسين توسيع الاستعلام

وجيه عبد الغني عبد الحسين
قسم علوم الحاسوب ، كلية العلوم ، جامعة بغداد، بغداد، العراق

الخلاصة

التطور الهائل في تقنيات المعلومات خصوصا في العقود القليلة الماضية ادى لزيادة في حجم بيانات الويب على شبكة الويب العالمية وهي تنمو بشكل كبير جدا. استرداد المعلومات ذات الصلة على الإنترنت أو أي مصدر بيانات باستخدام استعلام تم إنشاؤه بواسطة بضع كلمات اصبح تحديًا كبيرًا. لتجاوز هذا الأمر ، فإن توسيع الاستعلام (Query Expansion) له وظيفة مهمة في تحسين استرداد المعلومات (Information Retrieval)، حيث يتم إعادة إنشاء الاستعلام الأصلي للمستخدم إلى استعلام جديد من خلال إضافة

*Email: wajih_abdul_ghani@scbaghdad.edu.iq

مصطلحات جديدة ذات صلة وبنفس الأهمية. من مشاكل توسيع الاستعلام هو اختيار المصطلحات المناسبة. تؤدي هذه المشكلة إلى كيفية استرداد المستندات المهمة بدقة عالية واستدعاء عالٍ ومقياس Fعالي. في هذا البحث نقترح حل لهذه المشكلة من خلال تطبيق قياسات تشابه مختلفة باستخدام WordNet الإنجليزية. أثبتت النتائج التي تم الحصول عليها أنه باستخدام طريقة اختيار مناسبة ، يمكننا الاستفادة من WordNet الإنجليزية لتحسين كفاءة الاسترداد. العمل المقترح في هذه البحث هو استخلاص المصطلحات من جميع المستندات والاستعلام ، ثم تطبيق الخطوات التالية: المعالجة المسبقة ، وتوسيع الاستعلام بناءً على WordNetالإنجليزية ، وتحديد أفضل المصطلحات ، وزن المصطلحات ، وأخيرًا استخدام تشابه الجيب التمام (Cosine) وتشابه الجاكارد (Jaccard) للحصول على الوثائق ذات الصلة.

يتم تطبيق نتائجنا العملية على مجموعة بيانات DUC2002 التي تحتوي على 559 وثيقة موزعة على عدة فئات. متوسط دقة جيب التمام (للاستعلامات العشوائية) = 100٪ بينما متوسط دقة الجاكارد = 84.4٪ ومتوسط استدعاء جيب التمام = 86.8٪ بينما متوسط استدعاء الجاكارد = 73.4٪. متوسط قياس f لجيب التمام = 92٪ بينما متوسط قياس الجاكارد = 76٪.

1. Introduction

Information Retrieval (IR) is dealing with the retrieval and display of the information of interest. The user can arrive to the concerned information by the information retrieval system.

Usually, user's information is represented by means of a query. Therefore, many challenges might meet the IR system, one of which is the problem of vocabulary mismatch [1]. To treat this problem, the Automatic Query Expansion (AQE) was proposed by some researchers in the IR field. The goal of this technique is recreating the original query by appending new terms to it to obtain better results. Cui *et al*. classified the AQE techniques into two main classes: global analysis and local analysis [2].

The techniques of the *global analysis class* are independent from the main query or its result. In general, they use external knowledge sources to choose items for expansion, such as WordNet or thesaurus, whereas the *local analysis class* creates a new query depending on some retrieved documents of a previous search, for example relevance feedback [3].

Appending new terms to the main query can happen before either the primary search or the relevance-feedback search [4].

The IR System consists of three elements [5], namely the documentary database, the query subsystem, and the matching mechanism.

Refining the effectiveness of the information retrieval system depends on applying some techniques on it. One of these techniques is the query expansion [6].

The big data available in the Web has not been accompanied by techniques for retrieving the relevant data [7]. Usually, the search on web data does not produce relevant results because of four reasons; first, the words written by the user on the search engine are belonging to several topics, thus the results of the search do not give a clear result. Second, the shortness of the query may cause an ambiguity of what the user wants [8]. Third, the user does not know what he/she is searching for. Fourth, some users do not have the ability of formulating the suitable query [9].

2. Related Works

In 2006, Radwan *et al*. introduced a new function of fitness and compared their results with the genetic algorithm dependent on classical IR and cosine fitness function in the problem of query learning. Their function was applied to CISI, CACM and NPL. These three famous test collections were used to obtain a complete view of improving IR systems using genetic techniques [5].

In 2014, two methods of query expansion were proposed by Brandao. The first method is an unsupervised entity-oriented query expansion, which chooses terms expansion using taxonomic features innovated by the semantic structure. The second method includes techniques of machine learning so as to choose and rank the entities oriented for query expansion [10].

In 2014, Jain *et al*. suggested a technique that investigates the function of graph structure for query expansion and determines the significance of each node in the graph using WordNet. The most important nodes which represent the word senses were specified and appended to the original query [11].

In 2015, a method of query expansion for short queries on the Web was proposed by EI Ghali *et al*.. This method used the Latent Semantic Analyses (LSA) technique which is dependent on the context of the query. Three methods of query suggestion were used to extract the context from the search engine, namely the cosine similarity, the language models, and their fusion [12].

In 2018, Jabri *et al*. suggested a similarity measure using the query graph. This measure calculates the similarity between candidate terms and the initial query, text mining techniques, and explicit semantic analysis (ESA) measure [13].

The work proposed in this paper is extracting the terms from all the documents and the query, then applying the following steps: preprocessing, query expansion based on English WordNet, selecting the best terms, term weighting, and finally using the cosine similarity and Jaccard similarity to obtain the relevant documents.

3. Basic Concepts of Query Expansion System

The system of generating query expansion consists mainly of several steps. Next sections illustrate these steps.

3.1 Preprocessing

Preprocessing is a language dependent process. The main function of this step is to extract the character sequences form data set that increase user's original query, along with performing tokenization and linguistic preprocessing on them, while the same steps are applied to the user query.

3.2 Query Expansion

The technique of this step targets at appending additional related tokens to the main queries to improve the effectiveness of IR systems [6]. QE has an effective function in refining the information retrieval (IR), where the main query is updated to a new query by appending new related items with same importance.

There are several types of query expansion techniques, as summarized in Figure-1.

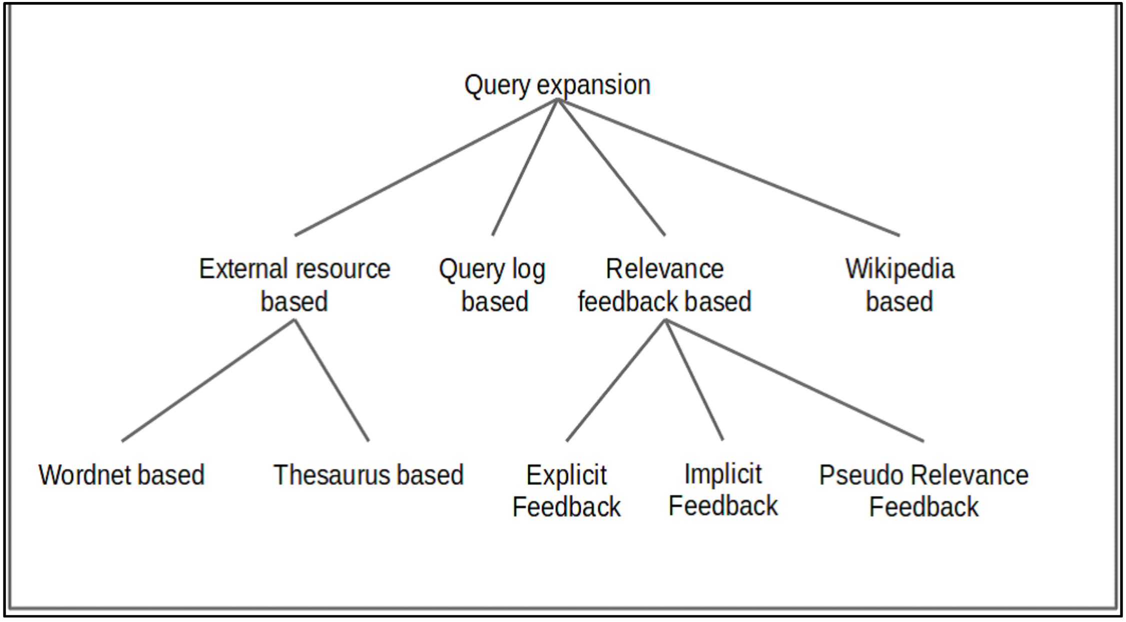


Figure 1-Types of Query Expansion

One of these techniques is based on WordNet which is used by this proposed work. WordNet is a lexical dictionary for several languages. The identical terms from several languages are linked by using synsets (set of senses). WordNet is used to get the equivalent terms in any language that verifies the user's information need. Hence, the synonyms' terms were added to the query.

Voorhees *et al*. [14] used WordNet for query expansion and reported negative results, where equivalent words were appended to the query. He noticed that this method produces a little difference in retrieval efficiency if the main query is formed very well. Smeaton *et al*. also used WordNet along with Point of Sale (POS) tagging for QE. The interesting point in this work is that it ignored the terms of the original query after the process of expansion [15].

3.3 Selection of the Best Terms of Query

In this step, choosing the best terms of query was done because the technique of query expansion makes more numbers of expansion tokens, but actually, these large tokens do not reflect the actual numbers of important tokens. Normally, a few numbers of expansion tokens are chosen since the effectiveness of the IR system becomes better when the expansion tokens are few. This selection is dependent on the existence, or not, of that term in the documents; if the original term exists in the documents then this term is assigned a weight one, and if the synonym term exists, it is assigned a half one.

3.4 Weighting and Ranking of Query Terms

In this important step of the system, ranks and weights of each query expansion tokens were calculated. In this step, the input is represented by the best terms of query selected from the previous step. The weight of tokens refers to the relevancy of tokens in the expanded query, which is then used in ranking the retrieved documents based on relevancy.

Term frequency- inverse document frequency (TF-IDF) is used in this paper as a weight measure of the individual tokens in both the expansion query and the data source.

TF-IDF measure is used to compute the weight of each item in the data source or in a query. This weight represents the importance of that item dependent on the number of times it appears in the documents.

To compute the weight of term (t) in a document (d), we must follow equation (1): [16]

$$W(t,d) = TF(t,d) \times \log\left(\frac{N}{DF(t)}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(1)}$$

where:

TF(t,d) is the occurrences count of term (t) in document (d). DF(t) is the documents count containing the term (t). N is the count of documents in the data source.

3.5 Similarity Measures

A similarity measure is the measure of how much alike are two objects. It can be used to calculate similarity between two queries, two documents, or one document and one query. The two measures which are used in this work are cosine similarity and Jaccard similarity.

3.5.1 Cosine Similarity Measure

The cosine similarity measure between any two data sets or two vectors is a measure that computes the cosine angle between them. This measure is used for orientation and not magnitude. It can be seen as a comparison between documents on a normalized space because the angle between documents is taken into consideration besides the magnitude of each word count (TF-IDF) of each document [17]. This measure is represented between (d1) and (d2) as shown in equation (2)

$$\text{Cosine\_similarity}(d_1, d_2) = \frac{\vec{V}(d1)\,.\,\vec{V}(d2)}{|\vec{V}(d1)|\,.\,|\vec{V}(d2)|} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(2)}$$

Figure-2 explains the vector space model for three sentences and three terms [17].

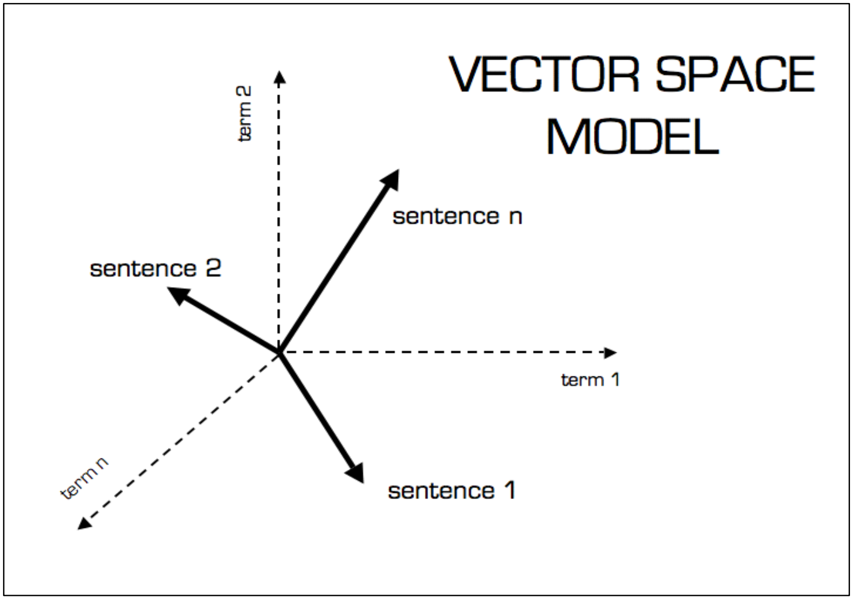


Figure 2- The Vector Space Model

3.5.2 Jaccard Similarity Measure

This measure is used to compute the similarity between two nominal attributes or between two sets by finding the intersection of these attributes or sets and dividing it by their union. Jaccard similarity between two sets A and B, the, i.e. JS(A, B), is represented as the size of their intersection divided by the size of their union. This is a very convenient measure as it is bounded between 0 and 1; JS(A, B) = 0 if and only if $A \cap B = \emptyset$, and JS(A, B) = 1 if and only if A = B. It has gained recent interest in its applications for finding documents (or web-pages) that are very similar but not the same, as well as in plagiarism detection. [18]
Mathematically, equation (3) clarifies the Jaccard measure.

$$\text{Jaccard\_Similarity (A , B)} = \frac{|A \cap B|}{|A \cup B|} \quad \ldots\ldots(3)$$

$$= \frac{|A \cap B|}{|A|+|B|-|A \cap B|}$$

1.6 The Evaluation of The Proposed System

The two most common measures for information retrieval performance are precision and recall [19]. The evaluation of the proposed system is necessary because it measures the performance of this system.
These measures are explained in equations 4 and 5.
Precision (*P*) is the fraction of retrieved documents that are relevant:

$$\text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \quad \ldots\ldots(4)$$

Recall (*R*) is the fraction of relevant documents that are retrieved:

$$\text{Recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|} \quad \ldots\ldots(5)$$

Besides the measures explained above, there is another measure which is the F measure; it is the weighted harmonic mean of precision and recall, as shown by equation (6).

$$\text{F-Measure} = 2 * \left(\frac{\text{precision . Recall}}{\text{Precision+Recall}} \right) \quad \ldots\ldots(6)$$

1. The Proposed Methodology

The process of the proposed system consists mainly of the following steps: preprocessing of data sources and query, query expansion depending on WordNet, term selection, term weighting, and ranking documents according to a score calculated through the Cosine and Jaccard similarity measures to obtain the relevant documents. Next sections illustrate the basic stages of this system.

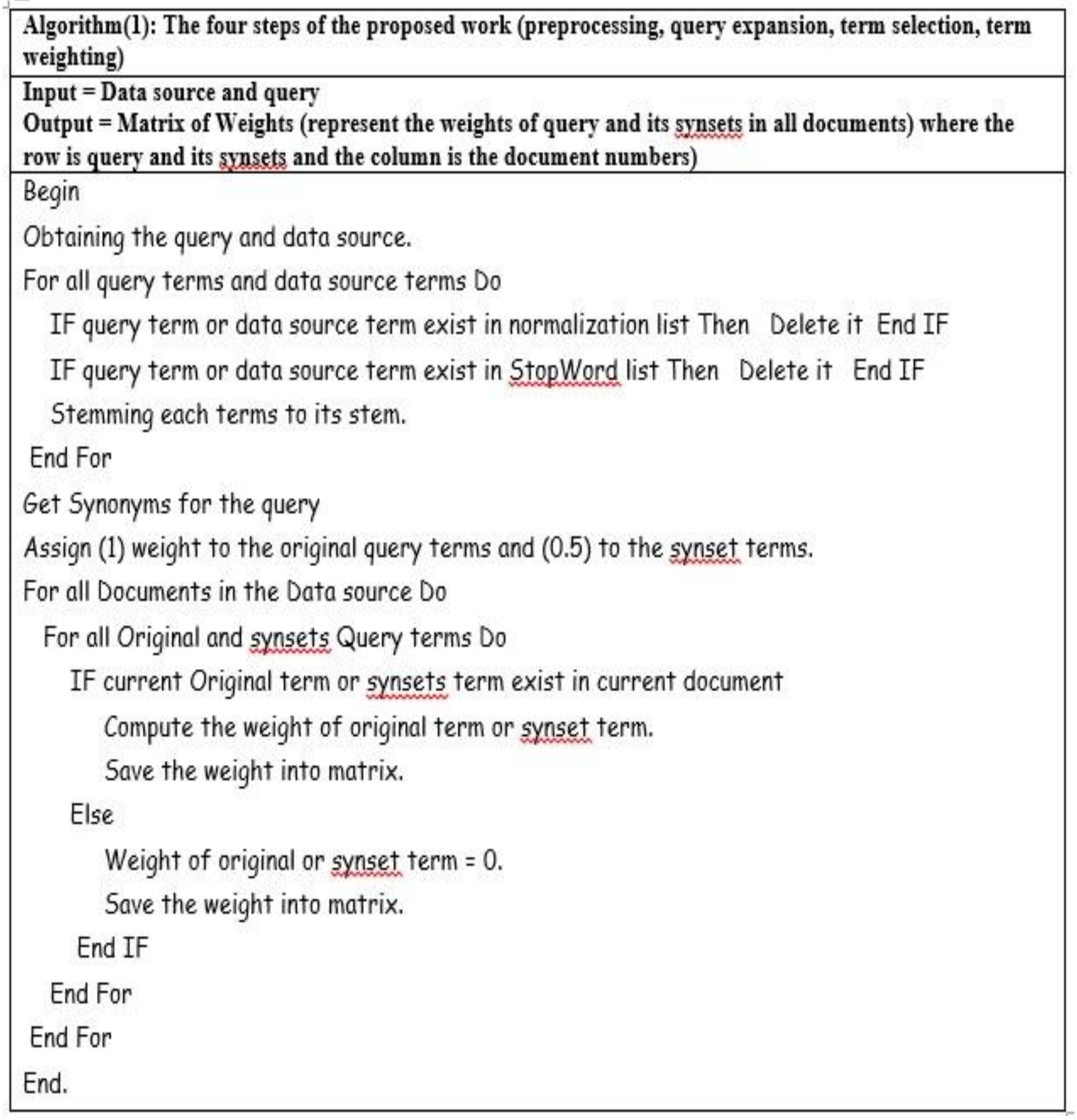
4.1 Query Preprocessing: this stage includes four steps (tokenization, normalization, stop words removal, and stemming)

- Extraction the text from the documents: extraction the entire texts from the documents and the query.
- Tokenization: the process of dividing the whole text into words.
- Removing stop words: removing the words which are used frequently like articles, adjectives, prepositions, etc.
- Word stemming: the procedure of restoring stems of the words.

4.2 Query Expansion: this step is concerned with finding the synonyms for the individual terms of a query. This operation is achieved by using the WordNet. Simply, this database contains, for each word in English language, its corresponding synonyms (synsets). Some words may have a lot of synonyms, thus a pruning operation is required to reduce them.

4.3 Synonym Selection: the aim of this step is to select a list of synonyms from the whole list of the synonyms for a specific term relying on the absence or presence of this synonym in the documents; if it is present in any of the documents in the collection, the synonym will be chosen, else it will be ignored.

4.4 Term Weighting: after all the previous steps, the term weighting step is responsible of the calculation of the weights of the remaining terms of a query by using TF-IDF weighting measure. Algorithm (1) shows the four previous steps.

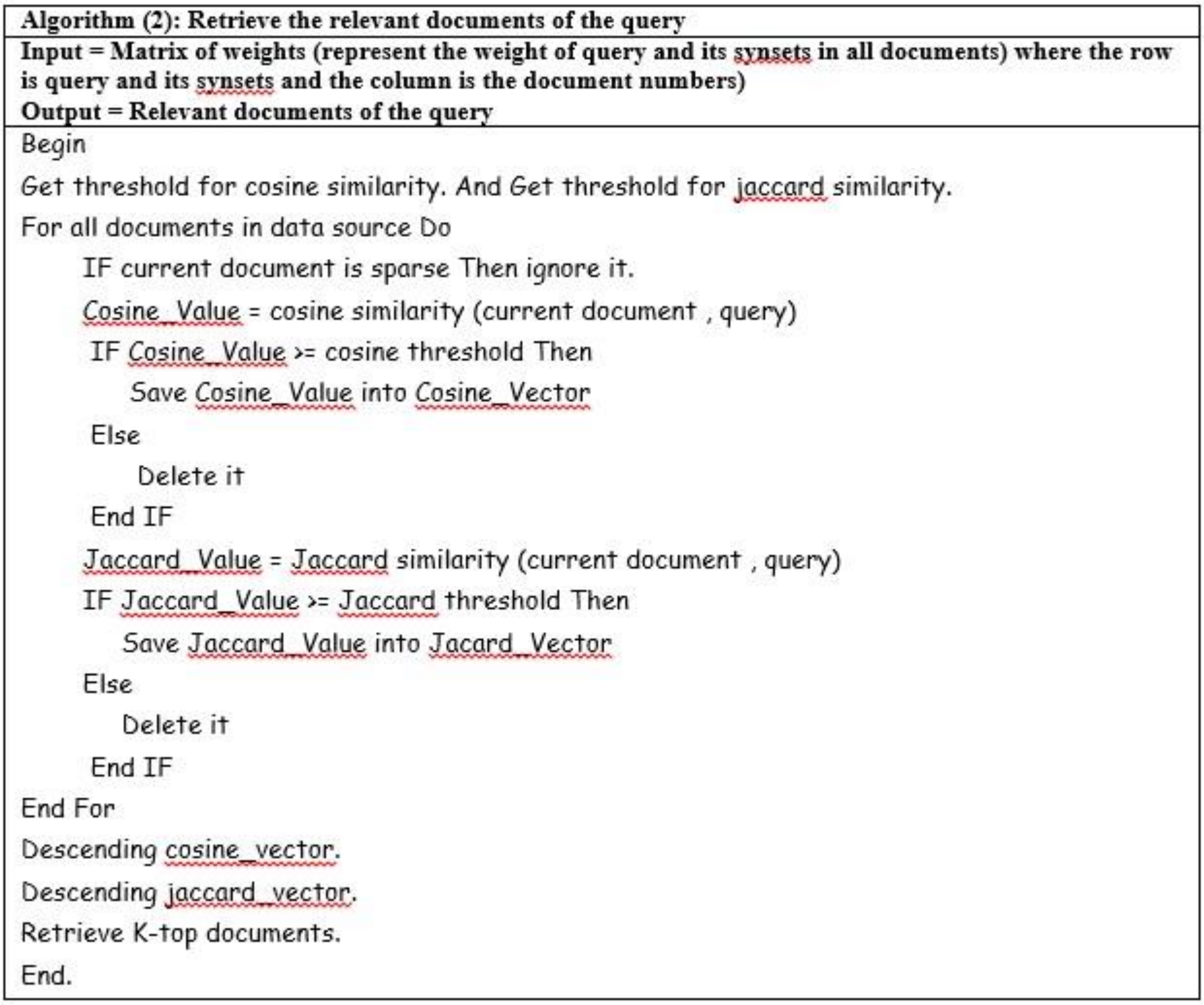


Algorithm(1): The four steps of the proposed work (preprocessing, query expansion, term selection, term weighting)

Input = Data source and query
Output = Matrix of Weights (represent the weights of query and its synsets in all documents) where the row is query and its synsets and the column is the document numbers)

```
Begin
Obtaining the query and data source.
For all query terms and data source terms Do
    IF query term or data source term exist in normalization list Then   Delete it  End IF
    IF query term or data source term exist in StopWord list Then   Delete it   End IF
    Stemming each terms to its stem.
 End For
Get Synonyms for the query
Assign (1) weight to the original query terms and (0.5) to the synset terms.
For all Documents in the Data source Do
   For all Original and synsets Query terms Do
      IF current Original term or synsets term exist in current document
          Compute the weight of original term or synset term.
          Save the weight into matrix.
      Else
          Weight of original or synset term = 0.
          Save the weight into matrix.
       End IF
    End For
 End For
End.
```

4.5 Calculating Similarity: this is the last stage in the proposed system where scoring was applied through two similarity measures, i.e. cosine similarity and Jaccard similarity, applied for each query document pair. Algorithm (2) demonstrates an algorithm to retrieve the relevant documents of the query.



Algorithm (2): Retrieve the relevant documents of the query

Input = Matrix of weights (represent the weight of query and its synsets in all documents) where the row is query and its synsets and the column is the document numbers)

Output = Relevant documents of the query

```
Begin
Get threshold for cosine similarity. And Get threshold for jaccard similarity.
For all documents in data source Do
        IF current document is sparse Then ignore it.
        Cosine_Value = cosine similarity (current document , query)
         IF Cosine_Value >= cosine threshold Then
              Save Cosine_Value into Cosine_Vector
         Else
               Delete it
         End IF
        Jaccard_Value = Jaccard similarity (current document , query)
        IF Jaccard_Value >= Jaccard threshold Then
            Save Jaccard_Value into Jacard_Vector
        Else
            Delete it
         End IF
End For
Descending cosine_vector.
Descending jaccard_vector.
Retrieve K-top documents.
End.
```

2. The Experimental Results

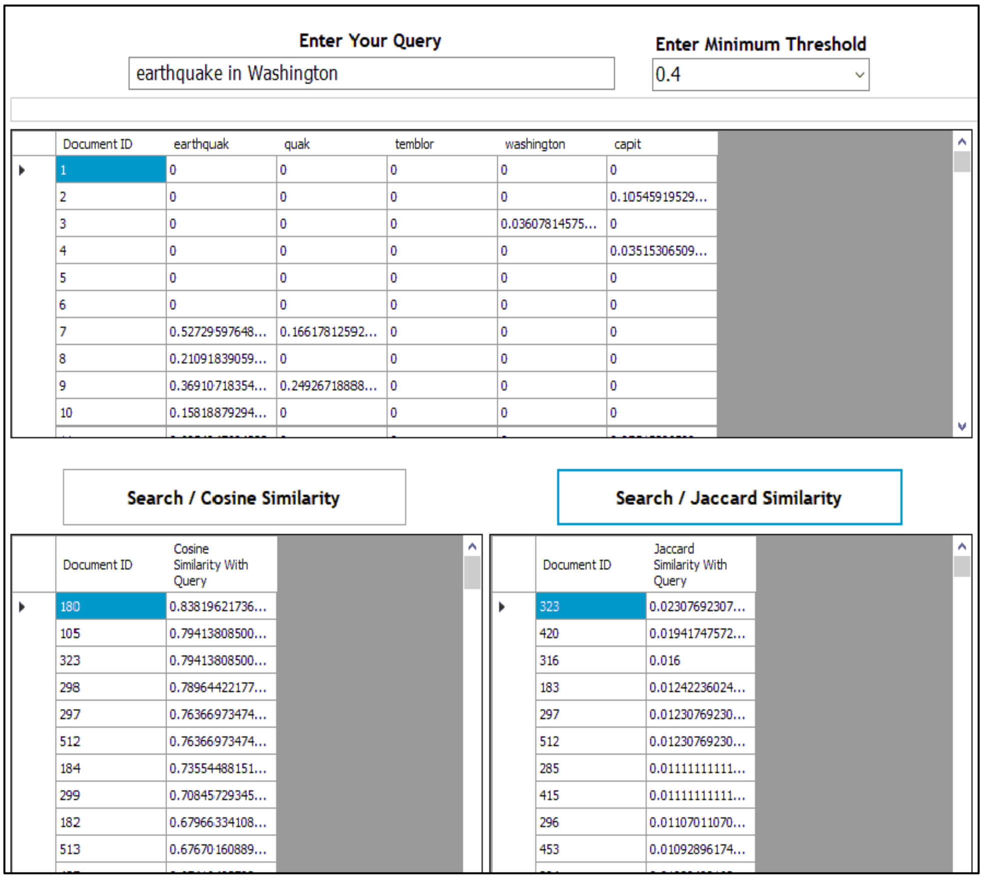
The proposed system used the summarized datasets (DUC2002) which is a free documents data source that contains 559 documents in multiple subjects such as Nature disasters, Politics, Middle-East, Sport, Health, etc.

This data source went through the four stages of preprocessing and then will be saved into text files to be ready for the next step.

This work uses C sharp or C# programming language, one of several languages that exist in visual studio 2015, as a tool for solving the problems of the practical part of paper. Also, it runs under windows 10 with 8 gigabytes of ram and core I5 (1.8) GH of intel cpu.

In the query side, we observed a query as a vector. The four preprocessing steps are applying on it, finding the synonyms depending on the WordNet, finding the best synsets, where the weight of each term is calculated by using TF-IDF, and finally, using the proposed system to measure similarities (cosine, Jaccard) to find the most retrieved documents.

As illustrated in Figure-3, the system reads a specific query which it searches for, along with the minimum threshold for the similarity measure value, and clicks on the search button.



Enter Your Query: earthquake in Washington

Enter Minimum Threshold: 0.4

Document ID	earthquak	quak	temblor	washington	capit
1	0	0	0	0	0
2	0	0	0	0	0.10545919529...
3	0	0	0	0.03607814575...	0
4	0	0	0	0	0.03515306509...
5	0	0	0	0	0
6	0	0	0	0	0
7	0.52729597648...	0.16617812592...	0	0	0
8	0.21091839059...	0	0	0	0
9	0.36910718354...	0.24926718888...	0	0	0
10	0.15818879294...	0	0	0	0

Document ID	Cosine Similarity With Query
180	0.83819621736...
105	0.79413808500...
323	0.79413808500...
298	0.78964422177...
297	0.76366973474...
512	0.76366973474...
184	0.73554488151...
299	0.70845729345...
182	0.67966334108...
513	0.67670160889...

Document ID	Jaccard Similarity With Query
323	0.02307692307...
420	0.01941747572...
316	0.016
183	0.01242236024...
297	0.01230769230...
512	0.01230769230...
285	0.01111111111...
415	0.01111111111...
296	0.01107011070...
453	0.01092896174...

Figure 3-The Results for “Earthquake in Washington“ Query

The search button will perform the following steps: Query Preprocessing, Query Expansion, Term Selection, Term Weighting, and Similarity Measure.

Because of the restricted area, this paper shows an example for one query and its result. Table-1 shows the TF-IDF weight for the first 20 documents when the system reads the query “earthquake in Washington“.

Table-1: TF-IDF Weights for “earthquake in Washington“ query

Doc_id items	earthquake	quake	temblor	Washington	capital
Doc1	0	0	0	0	0
Doc2	0	0	0	0	0.1054
Doc3	0	0	0	0.0360	0
Doc4	0	0	0	0	0.0351
Doc5	0	0	0	0	0
Doc6	0	0	0	0	0
Doc7	0.5272	0.1661	0	0	0
Doc8	0.2109	0	0	0	0
Doc9	0.3691	0.2492	0	0	0
Doc10	0.1581	0	0	0	0
Doc11	0.6854	0	0	0	0.0351
Doc12	0	0	0	0	0.0351
Doc13	0	0	0	0	0
Doc14	0	0	0	0	0.0351
Doc15	0	0	0	0	0
Doc16	0	0	0	0	0
Doc17	0	0	0	0	0
Doc18	0	0	0	0	0
Doc19	0	0	0	0	0.0351
Doc20	0	0	0	0	0

The proposed system assigns an assumption weight to the query terms; the original term will be assigned 1 and the synonyms will be assigned 0.5, as shown in Table-2.

Table 2-Assumption Weight for "earthquake in Washington" query

Query	earthquake	quake	temblor	Washington	capital
Proposed Weight	1	0.5	0.5	1	0.5

In the cosine similarity measure, document 180 is the document that had the top-scoring for this query, with a score of 0.8381962, whereas document 105 had a score of 0.7941380, and document 323 was the third with a score of 0.7941380. Whereas using Jaccard similarity, document 323 was the document with the top-scoring for this query, with a score of 0.0230769, whereas document 420 scored 0.0194174, and document 316 scored 0.016. Table-3 demonstrates the top ten scoring documents for the above query.

Table 3-Cosine and Jaccard Similarity Measures for "earthquake in Washington" query

Document_ID	**Cosine Similarity with Query**
Doc - 180	0.8381962
Doc - 105	0.7941380
Doc - 323	0.7941380
Doc - 298	0.7896442
Doc - 297	0.7636697
Doc - 512	0.7636697
Doc - 184	0.7355448
Doc - 299	0.7084572
Doc - 182	0.6796633
Doc - 513	0.6767016

Document_ID	**Jaccard Similarity with Query**
Doc - 323	0.0230769
Doc - 420	0.0194174
Doc - 316	0.016
Doc - 183	0.0124223
Doc - 297	0.0123076
Doc - 512	0.0123076
Doc - 285	0.0111111
Doc - 415	0.0111111
Doc - 296	0.0110701
Doc - 453	0.0109289

The proposed system was evaluated using precision, recall, and F1 evaluation measures, as explained in the above equations (4, 5 and 6, respectively). This system was applied on samples of the random queries. Table-4 illustrates the precision, recall and F1-measure of 5 random queries with minimum thresholds of 0.7 for cosine similarity and 0.003 for Jaccard similarity. It is worthy to note that these results are running on the first (50) documents.

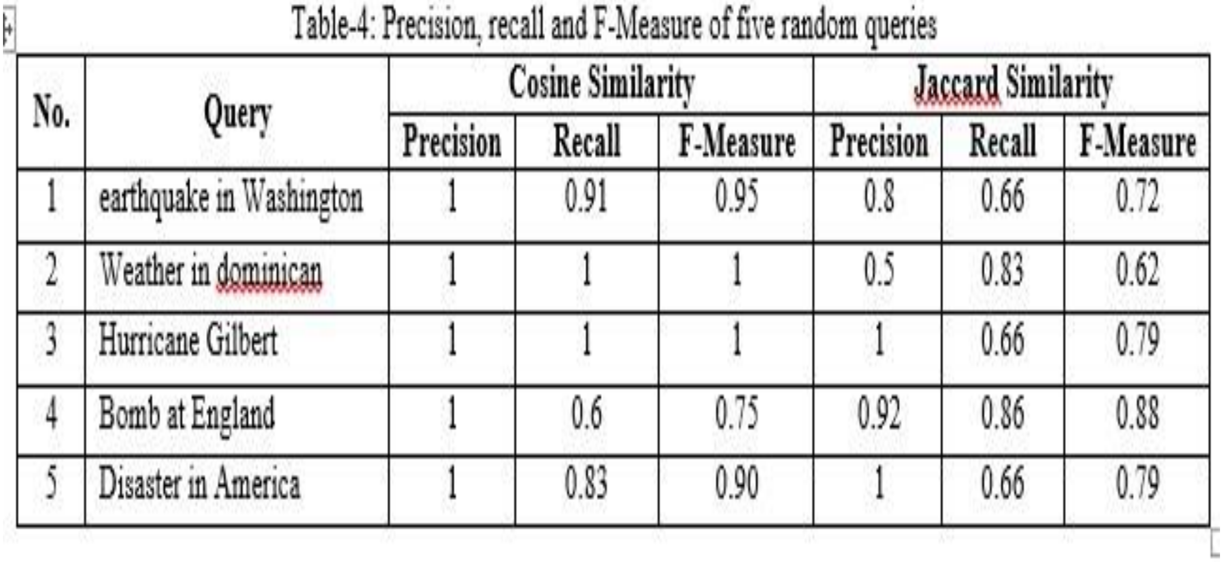


Table-4: Precision, recall and F-Measure of five random queries

No.	Query	Cosine Similarity			Jaccard Similarity		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
1	earthquake in Washington	1	0.91	0.95	0.8	0.66	0.72
2	Weather in dominican	1	1	1	0.5	0.83	0.62
3	Hurricane Gilbert	1	1	1	1	0.66	0.79
4	Bomb at England	1	0.6	0.75	0.92	0.86	0.88
5	Disaster in America	1	0.83	0.90	1	0.66	0.79

Conclusions and Future Work

Based on the results obtained from this work, a number of conclusions were obtained regarding the projected system.

- The query expansion is capable of overriding the problems of vocabulary mismatch in IR systems.
- The incompatibility between query items and document items highly affects the effectiveness of the retrieval operation.
- The precision and recall measures focus on the assessment on the retrieve of true positive documents. These measures will provide us with the percentages of the existing relevant documents and the false positives documents.
- The precision measure in the cosine similarity is better than that in Jaccard similarity because all the retrieved documents with it are relevant. On the contrary, in Jaccard similarity, not all the retrieved documents are relevant.
- Giving a high priority (high weight) to the original terms of the query will give much better results of similarity and evaluation measures.
- In the query expansion phase, the count of the synonyms for specific words may be large, and thus a pruning operation is required to reduce them.
- The obtained results confirmed that the cosine similarity measure is better than Jaccard similarity measure because its retrieved documents have more accuracy . We notice from the results that the average precision of cosine (for random queries) = 100%, whereas that of Jaccard = 84.4 %, and the average recall of cosine = 86.8% , while that of Jaccard = 73.4%. The average f-measure of cosine = 92% whereas the average f-measure of Jaccard = 76%.

In the future work, the use of genetic algorithms or any optimization algorithm in information retrieval may be better than the use of similarity measures because several related documents will be retrieved to the system in the genetic modification.

References

1. Carpineto, C. and Romano, G., **2012**. A survey of automatic query expansion in information retrieval, *Journal ACM Computing Surveys*, **44**(1), paper no. 1.
2. Cui, H., Wen, J. R., Nie, J. Y. and Ma, W. Y. **2002**. Probabilistic query expansion using query logs, Proceedings of the Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA, 7-11 May 2002, p 325-332. ACM.
3. Bilel, E., Ibrahim, B., Oussama B. K., Fabrice, E. and Narjès B. B. S. **2011**. Towards a Possibilistic Information Retrieval System Using Semantic Query Expansion. *International Journal of Intelligent Information Technologies*, **7**(4): 1-25.
4. Cuna, E. F., Alexander, M. R. and Peter, W. **1992**. Effectiveness of query expansion in ranked-output document retrieval systems, *Journal of Information Science*, **18**(2): 139-147.
5. Ahmed A. A. Radwan, Bahgat A. Abdel Latef, Abdel Mgeid A. Ali, and Osman A. Sadek. **2006**. Using Genetic Algorithm to Improve Information Retrieval Systems, *World Academy of Science, Engineering and Technology*, **17**(December): 6-12.
6. Ahmed Abbache et al., Arabic Query Expansion Using WordNet and Association Rules, University of Oran 1, Ahmed Ben Bella, Oran, Algeria.
7. Mikroyannidis, A. **2007**. : Toward a social semantic web. *Computer*, **40**(11): 113-115.
8. Spink, A., Wolfram, D., Jansen, M.B., **2001**. Saracevic, T.: Searching the web: The public and their queries. *Journal of the American society for information science and technology*, **52**(3): 226-234.
9. Broder, A. **2002**. A taxonomy of web search. In: *ACM Sigir forum*, **36**: 3-10. ACM.
10. W. C. Brandao. **2014**. Exploiting entities for query expansion. *In: Proc. of ACM SIGIR Forum*, Vol.48, No. 1, pp. 43-43.
11. A. Jain, K. Mittal and DK. Tayal. **2014**. Automatically incorporating context meaning for query expansion using graph connectivity measures, *Progress in Artificial Intelligence*, **2**(2-3): 129–139.
12. B. El Ghali, A. El Qadi, M. Ouadou and D. Aboutajdine. **2015**. Context-based query expansion method for short queries using latent semantic analyses. *In: Proc. of International Conference on Networked Systems*, pp. 468-473.
13. Siham Jabri, Azzeddine Dahbi, Taoufiq Gadi, Abdelhak Bassir. **2018**. Improving Retrieval Performance Based on Query Expansion with Wikipedia and Text Mining Technique, *International Journal of Intelligent Engineering and Systems*, **11**(4): 283-292.

14. E. M. Voorhees. **2005.** The trec robust retrieval track. In ACM SIGIR Forum, **39**: 11–20. ACM, 2005.

15. A. F. Smeaton, F. Kelledy, and R. O'Donnell. **1995**. Trec-4 experiments at dublin city university: Thresholding posting lists, *query expansion with wordnet and pos tagging of spanish*. Harman [6], pages 373–389.

16. https://www.onely.com/blog/what-is-tf-idf/ , at March **2020**.

17. http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/, at March **2020**.

18. Jacob Bank, Benjamin Cole. **2008**. Calculating the Jaccard Similarity Coefficient with Map Reduce for Entity Pairs in Wikipedia, Wikipedia Similarity Team.

19. Christopher D. Manning and Prabhakar Raghavan and Hinrich Schütze. **2009**. *An Introduction to Information Retrieval*, Cambridge University Press Cambridge, England.