ISSN: 0067-2904

Automatic Query Expansion for Arabic Text Retrieval

Alia Karim Abdul Hassan*, Mustafa Jasim Hadi
Department of Computer Science, Technology University, Baghdad, Iraq.

Abstract

Query expansion (QE) is a successful idea to overcome the weaknesses in the information retrieval performance. The QE requires finding out appropriate word synonyms of the query words in a process that can be made automatically without any user intervention. The candidate synonyms should be associated with an accurate meaning (sense) of the original word. Arabic language is rich in multiple meanings and this requires using the so-called word sense disambiguation (WSD). WSD in general is a task to discover the correct sense of a word within context. To disambiguate the word sense, three different traditional semantic measures are tested in this work; they are called *lch*, *wup*, and *path* respectively. The proposed system uses these measures along with an automatic synonym selection method employed to expand the query. The proposed system outperforms the traditional baseline system that has no query expansion technique in a rate from 10% to 18 % and reduces the latency in an approximate rate from 0.232 to 0.283 second for each query.

Keywords: Query Expansion; Arabic Information Retrieval; Word Sense Disambiguation; Artificial Bee Colony

توسيع الاستعلام التلقائي لاسترجاع النص العربي

علياء كريم عبد الحسن*، مصطفى جاسم هادي
قسم علوم الحاسبات، الجامعه التكنلوجيه، بغداد، العراق.

الخلاصة

توسيع الاستعلام (QE) هو فكرة ناجحة للتغلب على نقاط الضعف في أداء استرجاع المعلومات. يتطلب QE إيجاد مرادفات مناسبة لكلمات الاستعلام في عملية يمكن إجراؤها تلقائيا دون تدخل المستخدم. يجب أن ترتبط المرادفات المرشحة بمعنى دقيق (حس) للكلمة الأصلية. اللغة العربية غنية بمعاني متعددة وهذا يتطلب استخدام ما يسمى بإزالة الغموض عن المعنى (WSD). WSD بشكل عام هو مهمة لاكتشاف المعنى الصحيح للكلمة ضمن السياق. لإزالة الغموض عن معنى كلمة، يتم اختبار ثلاثة مقاييس دلالية تقليدية مختلفة في هذا العمل تدعى بـ *lch* ، *wup* ، و*path* على التوالي. يستخدم النظام المقترح هذه المقاييس جنبا إلى جنب مع طريقة اختيار المرادفات التلقائية المستخدمة لتوسيع الاستعلام. يتفوق النظام المقترح على نظام خط الأساس التقليدي الذي ليس لديه تقنية توسيع بمعدل من 10٪ إلى 18٪ ويقلل من التأخير بمعدل تقريبي من 0.232 إلى 0.283 ثانية لكل استعلام.

*Email: hassanalia2000@yahoo.com

1. Introduction

Information retrieval (IR) concerns in retrieving a subset of documents that satisfy the user's need from a collection of documents. Most of IR researches try to manipulate the textual documents that are written in the English language. In contrast, there are a few IR researches are concerned with Arabic language. This may be due primarily to the morphological complexity of Arabic language that makes it difficult to address the natural language processing (NLP) in general and the Arabic IR applications in special [1]. Another reason is the lack of publicly freely accessible Arabic test corpora [2]. Arabic IR may return a poor performance due to not retrieving the documents that have various derivatives or synonyms of the query words. In addition, the queries in IR systems are usually very short and it is difficult to solve the ambiguity along with finding the exact estimation needed by the user. Query expansion (QE) is a successful idea to overcome weaknesses in performance [3]. The QE requires finding out equivalent word alternatives (synonyms) for all or some of query words, this is made by either the user intervention, usually called interactive query expansion (IQE), or is made automatically without any user intervention and usually called automatic query expansion (AQE). As a note, since the IQE is outside the subject of paper, both the "automatic query expansion" and "query expansion" are used interchangeably to refer to the same thing in this paper. To avoid the wrong synonyms, the so-called word sense disambiguation (WSD) is needed. WSD in general is a task to discover the correct sense of a word within context. To disambiguate a word in a text, there is a need to an information resource. The most significant resources to solve the ambiguity are the structured resources, such as the machine-readable dictionaries, thesauri, ontologies, and WordNet. Arabic WordNet (AWN) is a free lexical resource for Arabic language based on the well-known Princeton WordNet (PWN) for English [4]. All the Arabic synonyms can be extracted from the AWN along with their senses, where each sense is associated to a set of synonyms. This paper resolves the ambiguity of words by using three common semantic measures, these are *lch*, *wup*, and *path* respectively. After disambiguating the senses of the words by any one of these measures, a set of synonyms of each winning sense is ready to filter again. The proposed system tries to automatically detect a best synonym of each word in the query, extracting their weights, adding them to the original query, and finally producing a set of relevant documents. The experimental results exhibit the performance superiority of the proposed system over the traditional system that has non-expanded query.

The rest of the paper is organized as follows: Section 2 introduces the important related works. Section 3 reviews in detail the traditional information retrieval mechanism. Section 4 shows a brief description of the WordNet-based semantic measures. Section 5 exhibits the mechanism of artificial bee colony algorithm. The remaining sections from 6 to 9 exhibit the proposed system, experimental results, discussion, and conclusions.

2. Related Works

A few works have attempted to address Arabic information retrieval (AIR) in general and Arabic query expansion in special. The most important works related to the proposed system are:

1- Mahgoub et al. [5] introduced a technique to address the semantic query expansion. The technique is based on a domain independent semantic ontology constructed from Arabic Wikipedia. They had three themes include the handling for the generalizations, morphological variants, concept matches, and synonyms with correct senses. The system is tested using Zad-Al-Ma'ad corpus. The comparison of their system against the traditional keyword based search is presented in their experiments.

2- Shaalan et al. [6] suggested a method for query expansion on AIR using Expectation Maximization (EM). EM distance is a major factor in the overall success of their system. Expanding queries in their work consist of three steps: Extracting top 10 documents, extracting top 100 keywords out of the top 10 documents, and eliminating irrelevant keywords using EM distance. The remaining words are then added to the original query to construct the expanded version of the query. The test data used is the INFILE test corpus from CLEF 2009.

3- Khafajeh et al. [7] designed and built automatic Arabic thesauri using term-to-term similarity and association techniques that can be used to improve the Arabic query expansion. Their system consisted of three integrated phases are the preparing documents, building a traditional AIR system, and building thesauri. The process of query expansion passes through three successive stages includes sending query items to thesaurus, get similarity items, and reformulation. Their work shows that the association-thesaurus has superior performance over the similarity-thesaurus. However, it has many limitations over the traditional AIR system in terms of recall and precision level. Experiments

conducted on a selected 242 Arabic abstract documents from the National Computer Conference and 59 Arabic queries.

4- Abbache et al. [8] attempted to study the impact of AWN on the performance of AIR. This study leads to propose an interactive query expansion (IQE) methodology. First, the user selects the appropriate word's part-of-speech in the original query, and then he/she reselects the appropriate synonyms. Experimental results show that the IQE strategy improves Mean Average Precision (MAP) by 12.6%, but no variant of automatic query expansion (AQE) strategies did. Finally, they reported that an appropriate use of Arabic WordNet as a source for AQE can improve effectiveness for AIR. The experiments are conducted on the Xnh-4500 corpus, which is a sub-corpus from the Xinhua collection contains 4500 documents.

3. Information Retrieval

Information retrieval (IR) is a process to represent, store, organize and search the information items. Information must be structured in some manner that ensures the relevant information retrieval [9].

3.1 Information Retrieval System Components

Basically, the IR system constitutes of three main components, these are [10]:

1- The documentary database: This component stores the documents and the representations of their information contents. It is associated with the indexer module. Textual document representation is typically based on index terms, which are the content identifiers of the documents.

2- The matching mechanism: It evaluates the degree to the document, which represents a satisfaction of the requirements expressed in the query and retrieves those documents that are judged to be relevant.

3- The query subsystem: It allows the users to formulate their information needs and presents the relevant documents retrieved by the system to them.

3.2 Extraction of Index Terms

Document preprocessing is a procedure divided mainly into four text operations [11], as follows:

1- Tokenize the raw text to extract the terms.

2- Lexical analysis of the terms with the objective of treating digits, hyphens, punctuation marks, case folding, and diacritics in Arabic language.

3- Elimination of stopwords with the objective of filtering out words with very low discrimination values for retrieval purposes.

4- Stemming of the remaining terms to allow the retrieval of documents containing syntactic variations of query terms. There are many available Arabic stemmers described in[12] such as Khoja, Light10, Berkeley Light, Al-Stem, SAFAR, and ISRI Arabic stemmers. While Khoja stemmer was well-known and widely used for NLP and IR applications, ISRI (Information Science Research Institute) stemmer is a newer and does not validate roots against a dictionary that make it more capable of stemming rare and new words [13].

3.3 Indexing Process

The fast query evaluation makes use of the so-called "inverted index". The inverted index maps terms to the documents that contain them in same as the index of a book that maps a set of selected terms to page numbers [14]. An inverted index consists of two major components[15]:

1- The search index: Each distinct term t has a count df_t of the documents that containing t and a pointer to the start of the corresponding inverted list.

2- The inverted lists: Each inverted list stores identifier for each document d that contains the term t and a frequency $tf_{d,t}$ of the term t in the document d.

Figure-1 illustrates a simple example of the inverted index data structure.

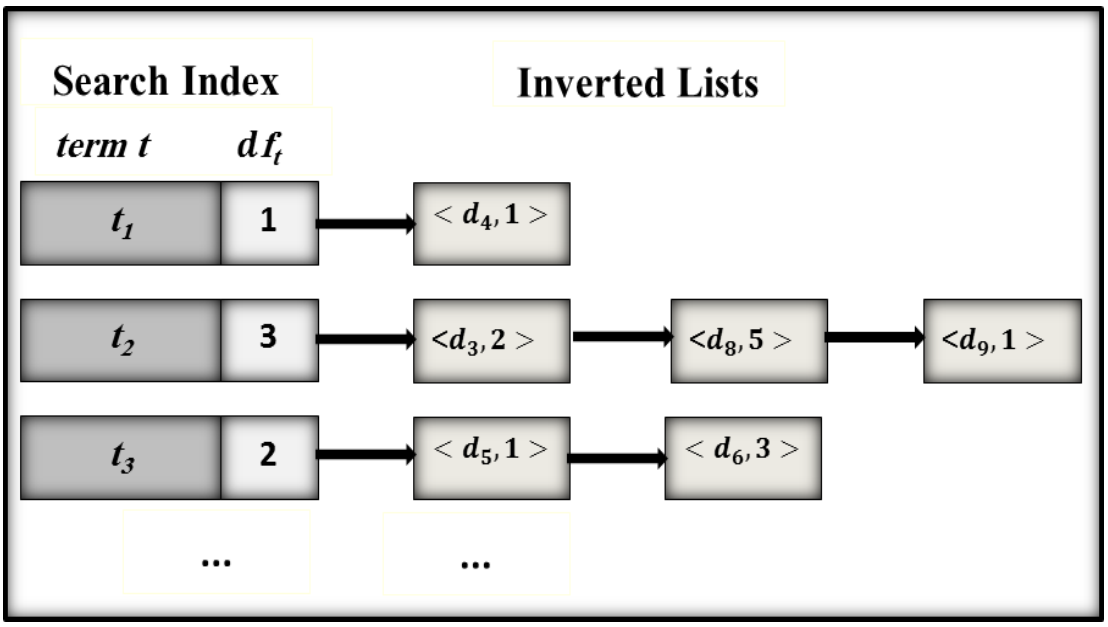

Figure1- Simple example of inverted index.

The term t_1 appears in one document $\{d_4\}$, therefore df_{t_1}=1, the occurrence in d_4 is just once, tf_{d_4,t_1}=1. Term t_2 occurs in three documents $\{d_3,d_8,d_9\}$, therefore df_{t_2} =3. The occurrence in d_3 is twice, tf_{d_3,t_2}=2 and in d_8 is five times, tf_{d_8,t_2} =5, and in d_9 is once, tf_{d_9,t_2} =1. Term t_3 appears in two documents $\{d_5,d_6\}$, therefore df_{t_3}=2, the occurrence in d_5 is just once, tf_{d_5,t_3}=1 while in d_6 is three times, tf_{d_6,t_3}=3 [14],[16], [17].

3.4 Traditional Search Approaches

Traditional search approaches for IR are the forward index search and inverted index search. The former depends on a forward index data structure while the latter depends on the inverted index data structure. Although it is more difficult to make and maintain the inverted index because the number of documents related to every term is changed dynamically, however, almost all the retrieval systems rely on the inverted index because its efficiency. In the forward index search (that maps from document identifier to content), all documents are visited and checked for the relevant, while in the inverted index search one can only care about documents that contain just the query terms[18].

3.5 Vector Space Model

Vector space model (VSM) allows to represent each document in a collection and each query as a point in a search space. The convergent points in this space are semantically similar and the diverging points are semantically distant. In VSM, the documents and queries are stored in weighted vectors. The weighted vector of a document d is defined as $\vec{d} = (w_{d,1}, w_{d,2}, \ldots, w_{d,n})$, while the weighted vector of a query q is defined as $\vec{q} = (w_{q,1}, w_{q,2}, \ldots, w_{q,n})$. The weight for a term in $\vec{d}$ or $\vec{q}$ is calculated by using *tf-idf* weighting scheme, tf refers to the term frequency and idf refers to the inverse document frequency. Cosine similarity metric is one of the most popular similarity measures that finds the normalized dot product of the two $\vec{q}$ and $\vec{d}$ vectors, as follows:

$$cos(q,d) = \frac{\vec{q}.\vec{d}}{|\vec{q}|.|\vec{d}|} = \frac{\sum_{t=1}^{n} w_{q,t}.w_{d,t}}{\sqrt{\sum_{t=1}^{n} w_{q,t}{}^2}.\sqrt{\sum_{t=1}^{n} w_{d,t}{}^2}} \quad (1)$$

Where $w_{q,t} = tf_{q,t} \times log\,(N/df_t)$ and $w_{d,t} = tf_{d,t} \times log\,(N/df_t)$. The expression $log\,(N/df_t)$ refers to *idf*, *N* is the number of documents in the document collection and df is the number of documents in which the term appears [9].

4. WordNet-Based Semantic Measures

Word sense disambiguation can be performed by using semantic measures based on WordNet. These measures have been grouped into four classes: path length-based measures, information content-based measures, feature-based measures, and hybrid measures [19]. This paper concerns with just the path length-based methods that include *lch*, *wup*, and *path* measures. The *lch* measure finds

the shortest path between two senses and scales the values by the maximum path length in the taxonomy graph. The *wup* measure finds the path length to the root node from the least common subsumer function of the two senses and scales the values by the sum of the path lengths from the individual sense to the root. The measure *path* is equal to the inverse of the shortest path length between two senses [20].

5. Artificial Bee Colony Algorithm

Artificial bee colony (ABC) is a meta-heuristic bio-inspired algorithm falls under the umbrella of swarm intelligence. It is introduced by Karaboga [21]. The main steps of the ABC algorithm are described as follows[22]:

Step 1: Initialize the population of random food sources and evaluate them.
Step 2: Produce new sources for the employed bees, evaluate them and apply the greedy selection process.
Step 3: Compute the probability values of the current sources to be used for the selection process by the onlookers.
Step 4: Produce new sources for the onlookers from selected sources, evaluate them and apply the greedy selection process.
Step 5: Determine the abandoned resources and send the scouts randomly in the search area for discovering alternative new food sources.
Step 6: Memorize the best food source achieved so far.
Step 7: If the termination condition is not met, return to step 2, otherwise stop.
In the initialization phase, the initial solutions are computed as follows[23]:

$$x_{ij} = x_{min\ j} + rand\ [0, 1] * (x_{max\ j} - x_{min\ j}) \quad (2)$$

Where $i \in \{1,..,N_S\}$, $j \in \{1,..,D\}$, N_S is the number of food sources, D is the number of optimized parameters, $x_{max\ j}$ and $x_{min\ j}$ refere to the upper and lower bounds for the dimension j. $rand[0, 1]$ is a random number between [0, 1]. In the employed bees and onlooker bees phases, the new solutions v_{ij} are computed as follows[23]:

$$v_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj}) \quad (3)$$

Where $i,k \in \{1,..,N_S\}$ and $j \in \{1,..,D\}$ are randomly chosen indexes, ϕ_{ij} is a random number between [-1, 1]. The selection of the new solutions in the onlooker bees depends on probability p_i that is computed as follows[23],[24]:

$$p_i = a \times \frac{f_i}{f_{max}} + b \quad (4)$$

Where (a+b=1), f_i is the fitness value of the food source i, f_{max} is the maximum fitness of food sources.

In the scout bees phase, a new source v_{ij} is randomly generated instead of an abandoned one depending on Eq. (2).

6. The Proposed AQE for AIR

The proposed system starts in applying one of traditional semantic measures, i.e either *lch*, *wup*, or *path*, for disambiguating the word sense for each query word to produce a best set of synonyms for each word. After that, it progresses to detect, in an automatic manner, a best synonym within the set for each word and adding them to original query with aim to retrieve the most relevant documents. The proposed approach is designed by using the artificial bee colony (ABC) algorithm. The original ABC algorithm is modified to suit the required task. The modified ABC is intended for two search processes, the first is to search the most relevant candidate document and the second is to search the best synonyms via using the corpus as a resource. More precisely, the proposed system passes two stages, in the first stage it infers the most relevant candidate document by using the original query and in the same time detects the best synonyms which have greater popularity than others in the corpus along with considering its neighboring words in the query. In the second stage, the system extracts first a set of documents that are similar to a large extent with the most relevant candidate and then applies the expanded query to these documents in order to produce the ranked list. Figure-2 shows the proposed system architecture.

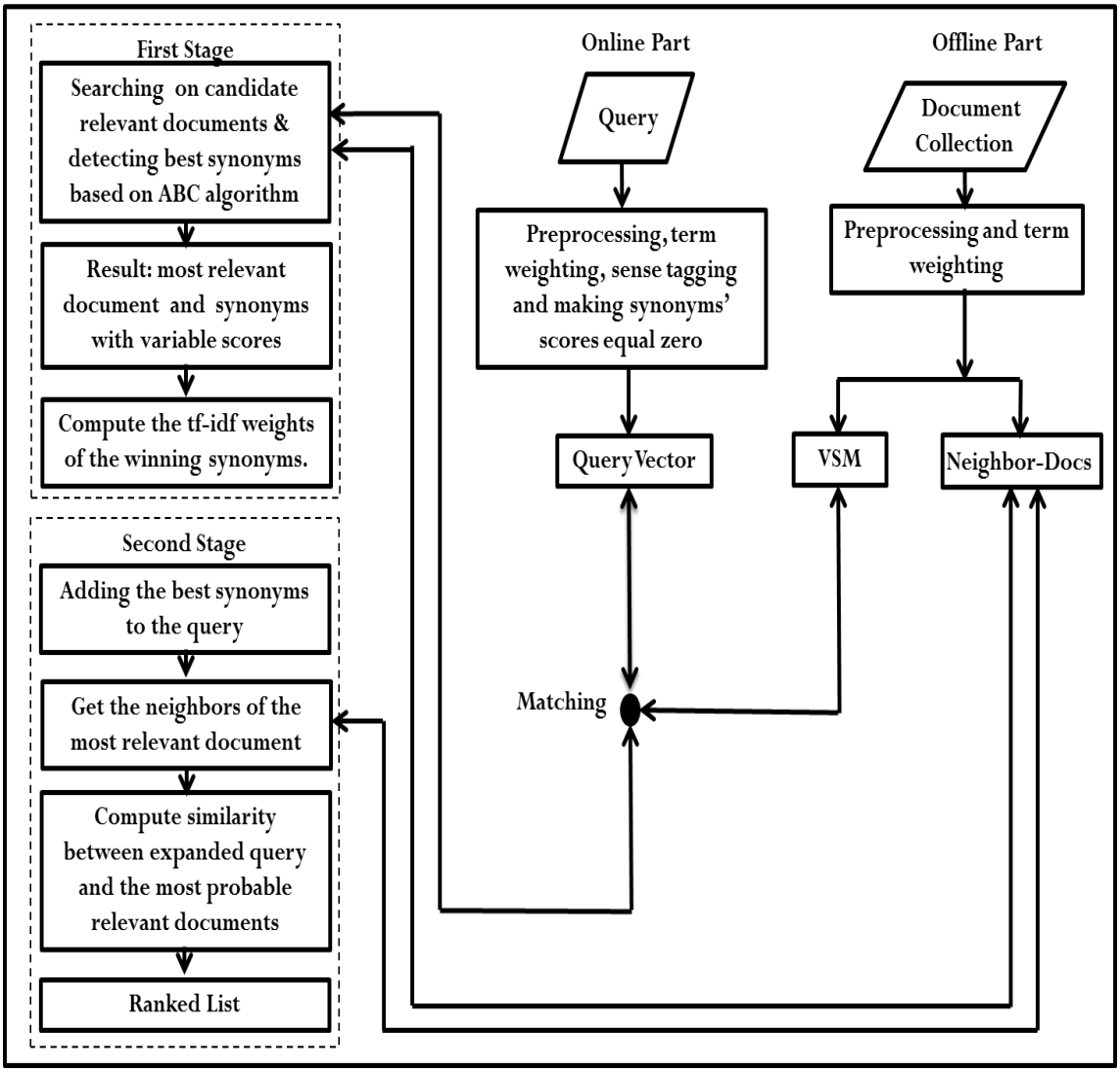

Figure 2- Proposed System Architecture

The first stage of the proposed system is an algorithm that depends on the original ABC algorithm described in an advanced section, it can be illustrated as follows:

Proposed System: First Stage
Input: D /* is a set of documents represented by a vector space model */, $Q = \{w_{q,i} \mid i = 1..n\}$ /* $w_{q,i}$ is a word associated with a *tf-idf* weight in a query Q*/, $S_q^{Synsets_{best}}$={$w_{q,i}^{Synset_{best}}$ |i = 1..n}: /* $w_{q,i}^{Synset_{best}}$ is the best sense detected by either *lch*, *wup*, or *path* measure. $Synset$ refers to a set of synonyms related to the best sense extracted from the AWN, their initial scores equal zero */ Output: doc^{best} /*the best relevant candidate document*/, $S_q^{synonym_{best}}$={$w_{q,i}^{synonym_{best}}$ |$i = 1..n$ } /*$w_{q,i}^{synonym_{best}}$ is the best synonym associated with its tf-idf weight of a word $w_{q,i}$*/. Begin /* Search Mechanism*/ *Step1:* Initialize the population: Set of random food sources (documents) selected from D. *Step2:* Calculate the fitness values of food sources in the population. Memorize doc^{global} known so far.

Step3: Update the synonyms scores of query words. Memorize the best synonyms known so far.
Step4: cycle = 1
Step5: Repeat
*5.1***:** Determine the neighbors $ni\_docs_i^{local}$ from *Neighbor-Docs* of the chosen food sources doc_i^{local} for the employed bees and evaluate them. Update the synonyms scores of query words.
*5.2***:** Determine the neighbors ni_docs^{global} from *Neighbor-Docs* of the chosen food sources doc_i^{local} based on the probability for the onlooker bees and evaluate them. Update the synonyms scores of query words.
*5.3***:** Produce new food sources for the abandoned food sources randomly from *D* using the scouts. Calculate their fitness values. Update the synonyms scores of query words.
*5.*4: Memorize doc^{global} and the best synonyms known so far.
*5.5***:** cycle = cycle + 1
*Step6***:** Until cycle = I_{max}
Step7: doc^{best}=doc^{global}, calculate the *tf-idf* weights for the best synonyms.
End.

Initially, the documents are randomly chosen in an integer interval from 1 to the document collection size. The fitness function in the algorithm is the cosine similarity between the query and documents. At each time the fitness is calculated, a synonym score is updated depending on the fitness, and the best synonym is memorized. The updating of the scores depends on the assumption that each document can be considered as a gloss (dictionary definition) of that synonym, and the query is the context where there is a need to find the suitable synonym. The synonym is then selected depending on the document that has the word that matches the query word synonym and offers the maximum cosine similarity with the query. The updating of the synonyms scores must be done after attending the matching between the synonyms and current document words as well as if the new fitness (cosine similarity) is larger than the old one.At each iteration, the new documents are evaluated and the synonyms scores are updated. The so-called *Neighbor-Docs* component is a data structure that has been offline constructed after the VSM has been achieved. *Neighbor-Docs* is simply a graph consists of ranked lists; each ranked list is produced by computing the cosine similarity between a document in the collection and all others in that collection. The only limit to this idea is that it may take a long time and space. However, this is performed offline and doesn't affect the AIR system efficiency. Also, it can be alleviated through using codes and resources that maximize the CPU utilization. The space can also be decreased by accepting only the similarities with a specific threshold. There are two types of neighboring documents, the local and global. The local neighboring documents (ni_docs^{local} in short) refer to the documents that are nearest to the current document (doc^{local} in short) within the population. The global neighboring documents (ni_docs^{global} in short) refer to the documents that are nearest to best document (doc^{global} in short) that is known so far within the population. In the employed phase, the current document is replaced with one of its neighbors. The neighbor is selected randomly from the documents that are very near to the current document doc^{local}, i.e. from ni_docs^{local}. To improve the diversity, the selected neighbor is again a subject to be replaced by another depending on a real random number within range in (0, 1). Thus, the selected neighbor can be altered by a new selection from its neighbors within *Neighbor-Docs*. In the onlooker phase, the neighbor's document is replaced with a document selected depending on Eq. (4). To get more diversity, the neighbors in the onlooker phase are selected randomly from the documents that are near to the global best document doc^{global} known so far, i.e. from ni_docs^{global}. In the scout phase, the abandoned documents are replaced with other documents selected randomly in the integer interval from 1 to the total document collection size. After the iterations reach to the desired limit, the algorithm output is the best document among others, doc^{global} , and a set of best synonyms, one for each query word. The second stage is illustrated as follows:

Proposed System: Second Stage
Input: doc^{best} ,$S_q^{Synonym_{best}}$. Output: Ranked list consist of the best relevant documents organized in descending order. Expanding and Matching Process: Begin *Step1:* Add the synonyms in $S_q^{Synonym_{best}}$ to query, call the result as $Q^{expanded}$. *Step2:* Add doc^{best} to its $neighbors$ list, call the result as $docs^{candidate}$. *Step3:* Calculate the cosine similarity between $Q^{expanded}$ and $docs^{candidate}$. *Step4:* Sort the results in descending order. End.

7. Experimental Results

The proposed system is experimented on ZAD corpus, a short term of Zad-Al-Ma'ad corpus that is written by the Islamic scholar "Ibn Al-Qyyim". ZAD corpus consists of 2730 Arabic documents, 25 Arabic queries, and supported by relevance judgments. For the IR systems that deal with large-scale databases, the retrieval efficiency becomes a critical task. In this paper the efficiency was in the position of interest. This paper tends to draw a simple simulation to a client-server database structure. The databases used in the proposed and traditional systems are stored in separate places within a server that includes the inverted index, VSM, and *Neighbor-Docs*. The consumed times for both traditional and proposed systems are recorded for each request to and import data from the server. The proposed system is run on a personal computer (Core-i5 @2.50 GHz, RAM 6GB, 64-bit operating system).

The experimental tests are compared to the traditional baseline system which has no any WSD mechanism. The proposed system tests three traditional semantic measures are *lch*, *wup*, and *path* for disambiguating all the senses in the query. Table-1 shows the tests for the first query in ZAD corpus, the query is "احكام صلاة الخوف".

Table 1-Performance Evaluation for the first ZAD Query

System	Ranked list (1'st ten docs in descending order)	Relevant documents (Human Judgment)	Intersection	Precisions, Recalls @10	Latency (in sec)
Traditional System (No AQE)	[1312, 3236, 3450, 1179, 2791, 2901, 1421, 2858, 3398, 2704]	[1151, 1312, 1313, 2729, 2858, 2859, 2901]	[1312, 2858, 2901]	0.3, 0.426	0.541
Proposed System (WSD based on *lch*, expanded query based on ABC)	[1312, 2690, 2729, 2858, 2901, 1151, 3815, 2791, 1313, 2701]	Same Judgment	[1312, 1313, 2729, 2858, 2901, 1151]	0.6, 0.857	0.840566
Proposed System (WSD based on *wup*, expanded query based on ABC)	[1312, 2690, 4000, 2741, 2729, 2858, 2901, 1151, 3815, 2791]	Same Judgment	[1312, 2729, 2858, 2901, 1151]	0.5, 0.714	1.102243
Proposed System (WSD based on *path*, expanded query based on ABC)	[1312, 1463, 2690, 2729, 2858, 2901, 1422, 1151, 3815, 2791]	Same Judgment	[1312, 2729, 2858, 2901, 1151]	0.5, 0.714	0.856191

Table -2 shows the average latency, the average of precision and recall @ 10, and the MAP measure for a sample of ten queries in ZAD corpus.

Table 2- Average performance evaluation with a ZAD collection of a sample of ten queries.

ZAD Average Performance	Traditional System (No AQE)	Proposed System (WSD based on *lch*, expanded query based on ABC)	Proposed System (WSD based on *wup*, expanded query based on ABC)	Proposed System (WSD based on *path*, expanded query based on ABC)
Avg. latency (sec./query)	0.970838	0.687959	0.726313	0.738945
Avg. precision	0.390000	0.590000	0.490000	0.550000
Avg. recall	0.234836	0.299646	0.276934	0.307353
MAP measure	0.581472	0.763023	0.684010	0.751566

Mean Average Precision (MAP) combines the recall and precision into a single score, it computes the precision in a gradually manner depending on the location of relevant documents in the ranked list. i.e., it considers that the first relevant document is more important than the second in the ranked list and the second is better than the third and so on. Depending on MAP measure entered in Table-2, the proposed system outperforms the traditional baseline system which has no WSD and no query expansion techniques in a rate of 18 % and reduces the latency in an approximate rate of 0.283 second for each query if the WSD method based on *lch* measure. If the WSD method based on *wup* measure, the proposed system outperforms in a rate of 10 % and reduces the latency in an approximate rate of 0.245 second for each query. If the WSD method based on *path* measure, the proposed system outperforms in a rate of 17% and reduces the latency in an approximate rate of 0.232 second for each query. Figure-3 shows the 11-point interpolated recall-precision curves of the traditional and proposed systems for the sample ten ZAD queries. The curves show the emergence sequence of the documents to the user where the points at the far left are better than others.

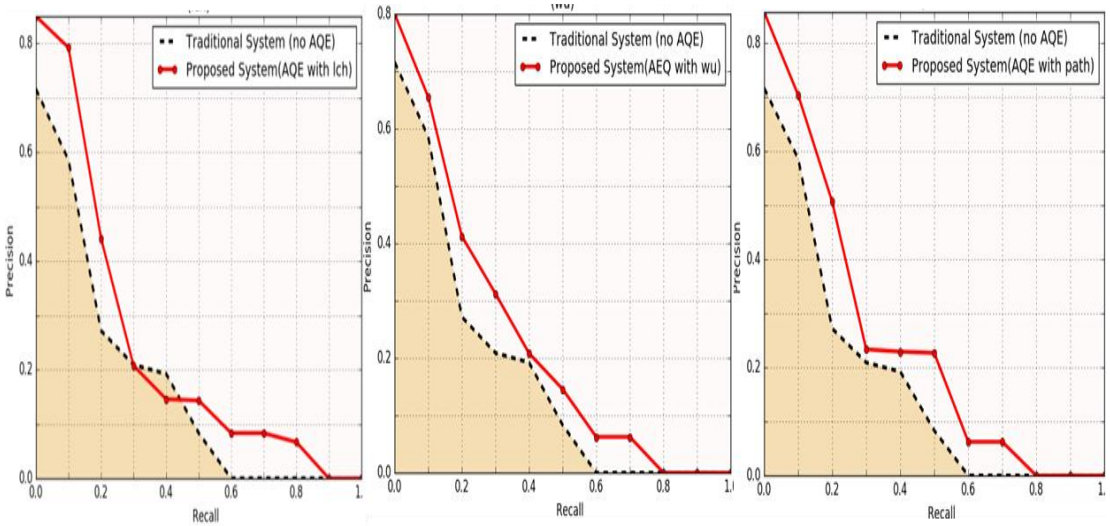

Figure 3- Average recall-precision curves for the sample of ten queries

8. Discussion

By viewing Table-2, the superiority in result quality is due to expand the queries and also to reach the documents that are prohibited through using the random search in the modified ABC algorithm. The proposed system succeeded in utilizing three different WordNet-based measures in order to disambiguate and expand the query words. In fact, it is very difficult to determine which measures are best due to the random search that conducted in the proposed system. However, through many trails, the results in Table-2 show the superiority of the measure *lch* due to own a high MAP. The superiority in MAP means producing a better ranked list. However, Figure-3 shows a drop in quality in one or

more queries when using *lch* measure. The dropping reveals that not all synonyms were selected correctly. Other measures *wu* and *path* have no drop in their curves. Since the results show that *path* measure has higher MAP than *wu* measure, it can be inferred that *path* measure is the preferred one among the rest to solve the problem of ambiguity. With regard to the average efficiency in Table-2, the high latency in the traditional system is resulted from using of the search by the inverted index. The matching of the query terms with the indexed words in the inverted index and extracting the documents' identifiers has a complexity to be increased incrementally with the number of existing indexed words. The proposed system has a direct search to documents in a pseudo-random manner that enables to get results in a reasonable response time. However, via viewing Table-1, the reader can wonder why the proposed system is inefficient for the given ZAD query? The answer to this question needs to understand the relationship between the corpus size, document size, and the distribution of terms. With regard to the document size, it was noticed through the experiments on ZAD corpus, that delay can be limited by number of words that travels within the document from the server to the client. The random search of the proposed system gives an advantage (in terms of chance) to getting different sizes of documents and reaching documents are prohibited by effect the weighting scheme and similarity function. This is one of reasons that the proposed system has superiority in the efficient in Table-2. Regarding to the corpus size and the distribution of terms, it can be said that the proposed system has an efficiency be directly proportional to the size of inverted index and sizes of inverted lists within inverted index. In fact, the proposed system will be inefficient if the inverted index is small size and there are numbers of query terms that match document terms and have low document frequencies. However, it is clear that in dealing with the large-scale databases, the direct access of the proposed system will be better than the index access in the traditional system. Also in large-scale environments, it is expected that the majority of the inverted lists will have larger sizes. As a result, the proposed system is very efficient if it deals with large-scale databases.

9. Conclusion

In this paper, it has been developed an automatic query expansion to improve the Arabic information retrieval. The proposed system passes two stages, in the first stage, it infers the most relevant candidate document by using the original query and in the same time it detects the best synonyms via using the corpus as a resource. In the second stage, the system extracts first a set of documents that are very similar to the most relevant candidate and then applies the expanded query to these documents in order to produce the ranked list. To avoid mistakes in word sense, the system applied three different traditional semantic measurers: *lch*, *wup*, and *path*, respectively. The experimental tests reveal that the *path* measure is the preferred one among the rest to solve the problem of ambiguity. Generally, the proposed system outperforms the traditional system in terms of the precision, recall, and the latency.

References

1. Larkey, L., Ballesteros, L. and Connell, M. **2007**. Light stemming for Arabic information retrieval. *Arab. Comput. Morphol.*, pp. 221–243.
2. Saad, M. K. and Ashour, W. **2010.** Arabic Morphological Tools for Text Mining. *6th Int. Conf. Electr. Comput. Syst.*, **18**: 19.
3. Rachidi, A., Bouzoubaa, T., ElMortaji, M., Boussouab, L., Bensaid, B., **2003**. Arabic user search Query correction and expansion. *Proc. COPSTIC'03*.
4. Elkateb, S., Black, W., Vossen, P., Farwell, D., Pease, A., and Fellbaum, C. 2006. Arabic WordNet and the Challenges of Arabic. *Proc. Arab. NLP/MT Conf.*, pp. 15–24.
5. Mahgoub, A., Rashwan, M., Raafat, H., Zahran, M,A. and Fayek, M.B. **2014**. Semantic Query Expansion for Arabic Information Retrieval. *EMNLP Arab. Nat. Lang. Process. Work.*, pp. 87–92.
6. Shaalan, K., Al-Sheikh, S. and Oroumchian, F. **2012.** Query expansion based-on similarity of terms for improving Arabic information retrieval. *Int. Conf. Intell. Inf. Process.*, pp. 167–176.
7. Khafajeh, H., Yousef, N., and Kanaan, G. **2010**. Automatic query expansion for Arabic text retrieval based on association and similarity thesaurus. *Proc. EMCIS*, 2010.
8. Abbache, A., Barigou, F., Belkredim, F.Z. and Belalem, G. **2014**. The Use of Arabic WordNet in Arabic Information Retrieval. *Int. J. Inf. Retr. Res.*, **4**(3): 54–65, 2014.
9. Baeza-Yates, R. and Ribiero-Neto, B. **1999**. *Modern Information Retrieval*. Addison

Wesley,Longman Publishing Co. Inc, 1999.
10. Radwan, A.A.A., Latef, B.A.A., Ali, A.M.A. and Sadek, O.A. **2008**. Using Genetic algorithm to improve information retrieval systems. *World Acad. Sci. Eng. Technol.*, **17**: 1021–1027.
11. Sugiyama, K. **2004.** Studies on Improving Retrieval Accuracy in Web Information Retrieval. Nara Institute of Science and Technology, 2004.
12. Dahab, M.Y., Ibrahim, A. and Al-Mutawa, R. **2015**. A Comparative Study on Arabic Stemmers. *Int. J. Comput. Appl.*, **125**(8): 38–47.
13. Ezzeldin A.M. and M. Shaheen, M. **2012**. A survey of Arabic question answering: challenges, tasks, approaches, tools, and future trends. *13th Int. Arab Conf. Inf. Technol.*, pp. 280–287.
14. Zobel, J. and Moffat, A. **2006**. Inverted files for text search engines. *ACM Comput. Surv.*, **38**(2): 1–56, 2006.
15. Abdullah H.S. and Hadi, M.J. **2014**. Improvement of Web Information Retrieval Using Swarm Intelligence. University of technology.
16. Lin J. and Dyer, C. **2010**. *Data-Intensive Text Processing with MapReduce*. Morgan and Claypool Publishers.
17. Manning, C.D., Raghavan, P. and Schutze, H. **2008**. *An Introduction to Information Retrieval*.
18. Jiang, S., Shang, M. and Deng, Z. **2011**. A design of the inverted index based on web document comprehending. *J. Comput.*, **6**(4): 664–670.
19. Meng, L., Huang, R. and Gu, J. **2013**. A Review of Semantic Similarity Measures in WordNet. *Int. J. Hybrid Inf. Technol.*, **6**(1): 1–12.
20. Pedersen, T., Patwardhan, S. and Michelizzi, J. **2004.** WordNet :: Similarity - Measuring the Relatedness of Concepts. *AAAI 2004, San Jose, CA*, pp. 1024– 1025.
21. Karaboga, D. **2005.** An idea based on honey bee swarm for numerical optimization. 2005.
22. Zhang, C., Ouyang, D. and Ning, J. **2010**. An artificial bee colony approach for clustering. *Expert Syst. Appl.*, **37**(7): 4761–4767.
23. Abdul-Hassan A.K. and Hadi, M.J. **2017**. Sense-Based Information Retrieval Using Fuzzy Logic and Swarm Intelligence. *Int. J. Multimed. Ubiquitous Eng.*, **12**(1): 363–376, 2017.
24. Tankasala, G.R. **2012**. Artificial Bee Colony Optimization for Economic Load Dispatch o f a Modern Power system. *Int. J. Sci. Eng. Res.*, **3**(1): 1–6.