Journal of Theoretical and Applied Information Technology
15th February 2017. Vol.95. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PROPOSED MABC-SDAIR ALGORITHM FOR SENSE-BASED DISTRIBUTED ARABIC INFORMATION RETRIEVAL

ALIA KARIM ABDUL HASSAN[1] , MUSTAFA JASIM HADI[2*]
[1]Computer Science Department, University of Technology/Baghdad, Iraq
[1] hassanalia2000@yahoo.com
[2] Computer Science Department, University of Technology/Baghdad, Iraq
[*]Corresponding author: [2] mustafa_awadi@yahoo.com, Baghdad/Al-Rusafa –10045.

ABSTRACT

Information Retrieval (IR) is the field of computer science that deals with the storage, searching, and retrieving of the documents that satisfy the user need. Distributed Arabic Information Retrieval (DAIR) is a model enables a user to access many searchable Arabic documents reside in different locations. DAIR is more complex than the centralized Arabic IR (AIR) because it requires addressing two significant additional problems that are the resource selection and the results merging. The Arabic language is a rich in multiple meanings (senses) in a lot of words and the tasking to find the appropriate meaning is a key demand in most of the AIR applications. This paper aims to improve the efficiency of the DAIR systems through using an algorithm belong to the swarm intelligence called Artificial Bee Colony (ABC) algorithm and to improve the result quality through using the query expansion. The MABC-SDAIR algorithm is the search approach used in this work. It aims to search the most relevant documents while at the same time it searches the best synonyms for the query expansion process. The experimental results exhibit the performance superiority of the proposed system over the traditional DAIR system that has a non-expanded query.

Keywords: *Arabic Information Retrieval; Distributed Arabic Information Retrieval; Artificial Bee Colony; Query Expansion.*

1. INTRODUCTION

Information retrieval (IR) is the science of retrieving a subset of documents that satisfy the user's need from a collection of documents. IR systems are used in a wide range of application areas such as web search, digital library search, social search, recommender system, etc. The major concern of IR is to find relevant information (documents) that satisfy the user need [1]. The growth of the Internet and digital libraries increased the need to address the IR in a distributed environment. The sources of information have become varied dramatically and their contents cannot be explored and indexed by a centralized IR system [2]. Distributed Information Retrieval (DIR) has received much attention in recent years. A simple DIR system is consisting of collection servers and a broker. When a user submits a query to the broker, the broker will propagate that query to a subset of collection servers that are carefully selected to satisfy the user need. After the query is processed by the selected collection servers, a ranked list from each one is returned to the broker. Finally, the broker merges the ranked lists into a unified ranked list and passing their contents to the user [3]. DIR is more complex than the centralized information retrieval (IR). It requires addressing two significant additional problems that are the resource selection (the decision to select the most appropriate databases to search) and the results merging (the unifying of the ranked lists, which were returned by the selected databases, into a single ranked list) [4]. Most of IR (or DIR) researches are concerned to manipulate the documents that are written in the English language. In contrast, there is a few IR (or DIR) researches are concerned with the Arabic language. This may be due primarily to the morphological complexity of the Arabic language that makes it difficult to address the Natural Language Processing (NLP) in general and the Arabic IR applications in special [5]. Also, the other important reasons are the lack of publicly freely accessible Arabic corpuses [5] and the lack of the efficient Arabic computational lexical resources.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In Arabic, there are a lot of words can be derived from a single word root. The IR for Arabic documents may return a poor performance if documents containing of the various derivatives of the query words are not retrieved. In addition, the queries in IR systems are usually very short and it is difficult to solve the ambiguity and find the exact estimation of user need. Query expansion is a successful idea to overcome the above problems [7]. This mechanism requires finding out the equivalent word alternatives (synonyms) from the correct senses of all or some of query words and inserting them into the query itself. The sense-based information retrieval systems aim to determine and using the suitable meanings for the words that have multiple meanings within their databases and/or queries to improve the retrieval quality. If the system is only interested to resolving the ambiguity in the query words then the synonyms of those suitable meanings are used for the query expansion purpose. The most used resources for resolving the ambiguity in the words is the external structured resources such as the dictionaries and thesauruses.

The external structured resources are used when the general senses of a word are available and there is no any information is known in advance about the sense of a particular word. Although these resources are a suitable to find the best senses and synonyms, but nevertheless they lack the domain-specific thesaurus relationships and also they require a lot of computation time and may be inconvenient in many IR systems. For this reason, an innovative technique that utilizes the corpus as an internal unstructured resource with a slight usage of the external structured resources becomes a desired task. This paper aims to improve the efficiency of the DAIR systems through using an algorithm belong to the swarm intelligence called Artificial Bee Colony (ABC) algorithm and to improve the result quality through using the query expansion. Artificial Bee Colony (ABC) is modified to suit the required purpose. While the modified ABC (MABC) explores the best relevant document it also simultaneously explores the best synonyms of query words that are initially extracted from an external structured resource called Arabic WordNet (AWN). AWN is a free lexical resource for Arabic language based on the well-known Princeton WordNet (PWN) for English [8].

The proposed system, using MABC-SDAIR algorithm, overcomes the problems in the research process that degrade traditional systems that rely on the inverted index. The high response time due to the indexed search, the decline in the results quality due to the resource selection, and the existence of the ambiguous words in the query are all addressed by our proposed system.

The rest of the paper is organized as follows. Section 2 reviews the brief description of the Arabic morphological analysis and what are the important Arabic stemmers used for the Arabic language. Section 3 introduces the concept of distributed information retrieval. Section 4 exhibits the mechanism of artificial bee colony algorithm. Section 5 reviews the important related works. The reminder sections 6 -10 exhibit the proposed MABC-SDAIR algorithm, experimental results, discussion of the results, conclusions, and future work respectively.

2. ARABIC MORPHOLOGICAL ANALYSIS

In Arabic language, the style of writing a letter varies depending on the place of the letter in the word. Three forms possible, the letter in a word may come at the beginning, middle or at the end. Moreover, there are proclitic, enclitic, prefix, and suffix letters can be added to the word [9]. Stopwords (common words) removal and also root-based stemming or light stemming improves the AIR performance. Root-based stemming removes the affix (prefix and suffix) and infix of a word, while the light stemming only removes the affix for the word. Light stemming for Arabic IR is believed to be better than the root-based stemming [10]. There are many available Arabic stemmers described in [11] 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 stemmer is a newer and a root-extraction stemmer without a root dictionary. In other words, ISRI (Information Science Research Institute) stemmer is similar to Khoja stemmer but without a root dictionary. Also, ISRI stemmer returns a normalized form for the unmodified (un-stemmed) words [11]. Saad in [10] uses a light stemmer for Arabic words inspired form ISRI Arabic stemmer. This light stemmer used to remove only prefix and suffix from words and doesn't convert words into their

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

root form as that done in the original ISRI Arabic stemmer. Also, the author presented a new reduction technique to increase of word matching chance is called morphAr. The basic idea of *morphAr* technique is to merge light stemming and rooting. If light stemming reduces the word form, then the light stem is returned, else, the root is returned. The proposed work in this paper depends on this technique, *morphAr*.

3. DISTRIBUTED ARABIC INFORMATION RETRIEVAL

Distributed Arabic Information Retrieval (DAIR) is a model in which a user accesses many searchable documents reside in a set of document collection servers. DAIR is more complex than the centralized Arabic Information Retrieval (AIR), it requires addressing two significant problems are the resource selection and the results merging.

Resource selection refers to finding a subset of document collections located on remote servers that have the most probability to contain the required information by a given query. This step is important to reduce the response time while trying to maintain the quality as much as possible [3]. When the broker receives a query, it forwards the query to the selected resources that use their local algorithms to rank the documents. Each selected resource returns a ranked list with scores to the broker. The broker normalizes the scores of the ranked lists for making them comparable in order to be unified in the next step, the results merging [12].

The results merging step is the last phase of the DAIR process. The individual ranked lists of the local collection servers are merged into a unified list that will be returned to the user submitting the query. Merging the ranked lists of individual AIR systems is a complex problem because the diversity of individual collection statistics that leads to inconsistent servers' scores [13]. From the perspective of the search engines, if there are different kinds of search engines are involved, the DAIR system should address the heterogeneity properties of the different search engines include collection statistics, retrieval strategy, document representation, indexing, and query representation [14]. The results merging phase is more important than the selection phase. This is because of its direct impact on the similarities and determines the relevant documents to the given query. If there is no effective merging, the retrieval quality will deteriorate even if we chose the most appropriate resources in the selection phase [15].

4. ARTIFICIAL BEE COLONY ALGORITHM

Artificial Bee Colony (ABC) algorithm is a stochastic-based bio-inspired algorithm falls under the umbrella of swarm intelligence. The ABC algorithm is introduced by Karaboga [16]. The main steps of the ABC algorithm are described as follows [17].

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:

$$x_{ij} = x_{min\ j} + rand\ [0,\ 1]^* (x_{max\ j} - x_{min\ j}) \quad (4.1)$$

Where i ϵ {1,..,N_S}, j ϵ {1,..,D}, N_S is the number of food sources, and 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:

$$v_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj}) \quad (4.2)$$

Where i,k ϵ {1,..,N_S} and j ϵ {1,..,D} are randomly chosen indexes, ϕ_{ij} is a random number between [-1, 1]. The selection of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

new solutions in the onlooker bees is depending on probability p_i that is computed as follows:

$$p_i = a \times \frac{f_i}{f_{max}} + b \quad (4.3)$$

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. (4.1) [18].

5. RELATED WORKS

Paltoglou in [13] offered a significant number of resource selection methods such as Glossary of Servers Server (GlOSS), Collection Retrieval Inference network (CORI), Cue Validity Variance (CVV), etc. The author shows that the CORI method is widely used as a baseline in the researches by a lot of authors and it is more effective than both GlOSS and CVV. CORI is an algorithm developed by Callan et al.[4,9] to calculate the collection servers' scores. It considers each collection server as a single giant document. Ranking the collection servers is similar to document ranking approaches used in the centralized IR systems [3]. It calculates belief scores of local collection servers based on a Bayesian inference network model with an adaptation of the Okapi term frequency normalization formula [20].

Many solutions for results merging issue are present in the literature. The simplest methods for addressing this problem are either the sequential combination of the results or the interleaved combination of the results in a Round Robin (RR) fashion. The RR merging method is used as a baseline for results merging by a lot of authors because of its simplicity and it introduces better results than the sequential combination method. Another solution that also serves as a baseline is the CORI merging method that is associated with the CORI resource selection method mentioned in advance. It is one of the most widely used methods due to its effectiveness and its simplicity [13]. Recently, Saoud and Kechid in [21] present an approach that exploits the social profile to improve both the source selection and the result merging process. To simulate a real DIR system and provide the social information, they construct their own document collection using a social bookmarking system and others from the Web. The reported results show this method improves the relevant metrics in DIR systems.

There are many intelligent methods inspired of swarm intelligence techniques were previously applied to the centralized Engligh IR systems. Ramya and Shreedhara in [22] offered in their paper a brief review on the application of swarm intelligence to centralized IR systems. They offer different swarm intelligence methods that aim to improve the search mechanism in the large-scale databases and the Web. Drias and Mosteghanemi in [23] designed an algorithm called BSO-IR inspired from the Bees Swarm Optimization (BSO) algorithm to explore the prohibitive number of English documents to obtain the most relevant results. They used CACM and RCV1 corpuses to test their experiments. The our previous work in [24] published recently presents a new technique called WSD-IR that uses the Artificial Bee Colony (ABC) approach to address the problem of using the Word sense Disambiguation (WSD) in the centralized English IR systems. The original ABC algorithm was modified to suit the required solution. The WSD-IR technique had outperformed the traditional technique in the terms of the latency and solution quality. CACM and NPL corpuses were used for conducting the experiments. The work was compared with the traditional inverted index approach.

A few works have attempted to address the AIR (or DAIR) in general and the Arabic query expansion in special. Mahgoub et al. in [25] 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. Shaalan et al. in [26] suggest a method for query expansion on the 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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

query to construct the expanded version of the query. The test data used is the INFILE test corpus from CLEF 2009. Khafajeh et al. in [27] 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 similar 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 document from the National Computer Conference and 59 Arabic queries.

6. PROPOSED MABC-SDAIR ALGORITHM

In the distributed information retrieval systems, the resource selection is a mechanism to improve efficiency while the results merging is a mechanism to improve result quality. However, many challenges related to the response time and result quality are still under discussion among a lot of authors. Although the resource selection mechanism improves efficiency, but it may harm the result quality. Also, the results merging mechanism improves result quality, but it may harm the system efficiency due to the complex online computations. This is why we need an innovative tool to overcome the weaknesses in the efficiency and the result quality. The proposed system tries to address these two problems using an algorithm inspired from the swarm intelligence optimization field. The MABC-SDAIR algorithm is the search approach used in this work. It is inspired from the original ABC algorithm that is described in advance. MABC-SDAIR algorithm aims to search the most relevant documents while at the same time it searches the best synonyms for the query expansion process. In other words, we need the stochastic optimization search of MABC-SDAIR algorithm to increase the efficiency while we need the query expansion to increase the result quality.

In this paper the traditional algorithms for the resource selection and results merging is used for the comparison process. The CORI methods for both the resource selection and merging are used for the traditional DAIR system. Whereas the selection algorithm in our proposed system is conducted randomly using the MABC-SDAIR algorithm and the merging method is simply the RR merging method. The proposed system consists of three main components as follows:

1. The documentary database (offline part): Let D is a set of document collections used in the proposed system, where D_i $(1 \leq i \leq n)$. Each collection is stored on a server S_i as local Vector Space Model (VSM) after the preprocessing and the term weighting have been achieved. Also, there is an additional server used to store the neighbor-docs data structure.
2. The query vector (online part): Let q is a query submitted to the DAIR system. The query should convert to query vector after the preprocessing and the term weighting have been achieved. Also associate each query word with list of different synonyms, one synonym for each sense, the synonyms are extracted from the Arabic WordNet. Make the synonyms weights equal to zero.
3. The matching mechanism (MABC-SDAIR algorithm) is illustrated below:

- Initialize the population: Set of random food sources (documents) scattered in all the servers S_i $(1 \leq i \leq n)$.
- Calculate the fitness values of food sources in the population. Update the synonyms weights of query words.
- cycle = 1
- Repeat
  - Memorize the locals and global best so far solutions. Let $Lbest\_doc_i$ is the best so far solution for a server S_i, and $Gbest\_doc$ is the global best so far solution.
  - Determine the neighbors from the $Lbest\_doc_i$ neighbor-docs of the chosen food sources for the employed bees and evaluate them. Update the synonyms weights of query words.
  - Determine the neighbors from the $Gbest\_doc$ neighbor-docs of the chosen food sources based on the probability for the onlooker bees and evaluate them. Update the synonyms weights of query words.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Produce the new food sources for the abandoned food sources by the scouts after scattering them to all the servers. Calculate their fitness values. Update the synonyms weights of query words.
- cycle = cycle + 1
- Until cycle = I_{max}
- Perform the query expansion by adding the synonyms that have the largest synonyms weights.
- Evaluate (using the expanded query) each local best solution with its associated neighbor-docs to produce a ranked list for each server S_i.
- Merge all the ranked lists using the RR merging method.

The main factor to success the MABC-SDAIR algorithm described above is the using of neighbor-docs data structure that is constructed offline using the $\varepsilon-$neighborhood document similarity graph G^{ε} after the Vector Space Model (VSM) has been completed. Given a vector space model $D_{n\times m}$ for a document collection. Let $G = (V, E, W)$ is an undirected weighted graph for a document collection in which V is a set of documents $d_i \in D$, E is a set of edges refer to the document relationships, and W is the similarity weights. For each pair of documents d_i and d_j, there is an edge $e_{ij} \in E$ connects the respective documents with weight w_{ij} equal to the cosine similarity. In order for the edges with low weights to be not included in the similarity graph, a threshold ε should be used. The threshold ε is a parameter to control the number of links in the graph. The higher the value of ε, the fewer the links can exist in the graph with high similarity scores. In other words, the $\varepsilon-$neighborhood document similarity graph G^{ε} is the document similarity graph $G = (V, E, W)$ in which each edge e_{ij} in the graph can connect two nodes (documents) with weight w_{ij} represents the cosine similarity result for the two nodes d_i and d_j if and only if the cosine similarity result $\geq \varepsilon$. The proposed system constructs one similarity graph G^{ε} for all the document collections in the system and preserves it in a single server for improving the neighboring search in the MABC-SDAIR algorithm at the query time. For the simplicity, we called all the documents connected with a specific document (one node in the graph) by name the "neighbor-docs". The neighbor-docs, then, is a data structure constructed offline after the global Vector Space Model (VSM) has been completed. It is simply a ranked list of documents that are associated with a corresponding document in the collection as if the document is a query. The neighbor-docs data structure has a major role in feeding the proposed system with the aim to find the optimal solutions.

Initially the documents are randomly chosen in an integer interval from 1 to the total collections size (i.e. summation of sizes of all the collections). The fitness function in the algorithm is the inner product between the corresponding weighted term vectors that computes the similarity between a query and a document. At each time the fitness is calculated, the synonyms weights are updated depending on Eq. (6.1) as follows:

$$S\_weight = \left(0.2 * w_d + 0.8 * \sum w_q . w_d\right) / \max tf_q \quad (6.1)$$

Where the expression $\sum w_q . w_d$ refers to the similarity between the query and the current document. This expression is used for updating the weight depending on the assumption that each document can be became a gloss or a definition and the query is the context where we need to find the correct word meaning (or the suitable synonym). The synonym is then selected depending on the document that has the word match the query word synonym and offers the maximum similarity with the query. However, to alleviate the side effect of the expression value, with respect to the other weights in the other query words, the system enters the weight of the word in a document that match the query word synonym as a factor in the equation. The equation also considers the query word normalization by dividing on the maximum term frequency within the query. An update of the synonyms weights must be done after the existence of matching between the synonyms and current document words as well as if the new weights resulting from Eq. (6.1) are larger than the old weights.

At each iteration, the new documents are evaluated and the synonyms weights are updated. The document has the best value at all, Gbest_doc, and the document has the best value within a local server S_i, $Lbest\_doc_i$, are all memorized. In the employed phase, the current document is replaced with one of its neighbors. The neighbor is selected randomly from the documents that are near to the current document,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

i.e. from its neighbor-docs. In the onlooker phase, the neighbor's document is replaced with a document selected depending on Eq. (4.3). To get more diversity, the neighbors in the onlooker phase are selected randomly from the documents that are near to the global best document so far, i.e. from the neighbor-docs of Gbest_doc. In the scout phase, the abandoned documents are replaced with other documents selected randomly in the integer interval from 1 to the total collections size. After the iterations reach to the desired limit, we gain the best document $Lbest\_doc_i$ for each server and also synonyms for each query word with variable weights. The final steps of the algorithm are to determine the best synonyms for query expansion process and then to obtain the ranked lists and merge them into a single ranked list. The best synonyms are determined simply by taking the largest synonyms weights. After we select the best synonyms, we expand the query by adding these synonyms to the query. Each ranked list is constructed using the similarity between the expanded query and the documents from a list consists of $Lbest\_doc_i$ and its neighbor-docs. The merging process is achieved using the RR merging method.

7. EXPERIMENTAL RESULTS

The proposed system is experimented on two different corpuses. The first is Zad-Al-Ma'ad corpus, namely ZAD for short (2730 Arabic documents, 25 Arabic queries, supported by relevance judgments), that is written by the Islamic scholar "Ibn Al-Qyyim". The second is NLEL Arabic Wikipedia corpus, namely NLEL for short (11638 Arabic Wikipedia documents in SGML format, 193 queries/questions)*, created by Benajiba and others [28] and published from NLEL of the University of Valencia [29]. Since there is no relevance judgments are available for the NLEL queries, we have made our relevance judgments automatically for each query. This is done by selecting a random sample with a random size from each similarity list for each query. This similarity list is resulted from using the inverted index-based search and consists of all relevant documents (that exceed a specific threshold) for a query.

For applying the distribution concept, the ZAD document collection is shuffled and divided into four parts and each one is located on a distinct server. Each sub-collection has a different size and the range for each one on a server S_i is as follows:

Server S_1: [1..540] , server S_2:[541..1180], server S_3: [1181..1920], server S_4:[1921..2730].

Also, we divide the NLEL document collection after the shuffling process into four parts with ranges as follows:

Server S_1:[1..1500] , server S_2:[1501..4000), server S_3:[4001..7500], server S_4:[7501..11638].

The experimental tests focused on the comparison between the proposed MABC-SDAIR algorithm and the traditional DAIR algorithm that has a non-expanded query. The given query is transformed into query vectors, each one corresponding to a local Vector Space Model (VAM) located on a remote server. The traditional algorithm selects a subset of servers that have most probable relevant documents, searches the local inverted indexes in the corresponding servers, and then achieve the query-document similarity to find the corresponding ranked lists. After that it achieves the results merging for getting a unified result. The selection and merging processes are performed using CORI methods. The MABC-SDAIR algorithm searches in a pseudo-random manner for the documents on the servers locally in a specific server or globally in all the servers according to MABC-SDAIR structure. To highlight the comparison between the proposed search and the traditional search, a sample of the first ten queries is selected from each corpus to evaluate the results. Tables 1 and 2 show the relevant documents within the rank 10 in the ZAD and NLEL collections respectively using the traditional and proposed algorithms. Traditional algorithm is tested once on two selected servers and again on three selected servers out of four servers. The two tables show the average evaluations of the performance with respect to the first ten queries of ZAD and NLEL collections respectively. Fig. 1 and 2 show the 11-point interpolated recall-precision curves for the traditional and proposed algorithms.

The curves constructed using the average of precision and recall at rank 10 of the sample queries.

* http://users.dsic.upv.es/grupos/nle/?file=kop4.php

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1. Average Performance Evaluation Of The ZAD Collection For The First Ten Queries.

Average performance for ZAD collection	Traditional Algo. (2 selected servers)	Traditional Algo. (3 selected servers)	MABC-SDAIR Algo.
Average of documents that are visited for each query out of (2730) documents.	270	389	533
Average of latency (Sec. /Query).	0.114864	0.162772	0.132972
Average of precision.	0.260000	0.310000	0.370000
Average of recall.	0.176110	0.220241	0.306096

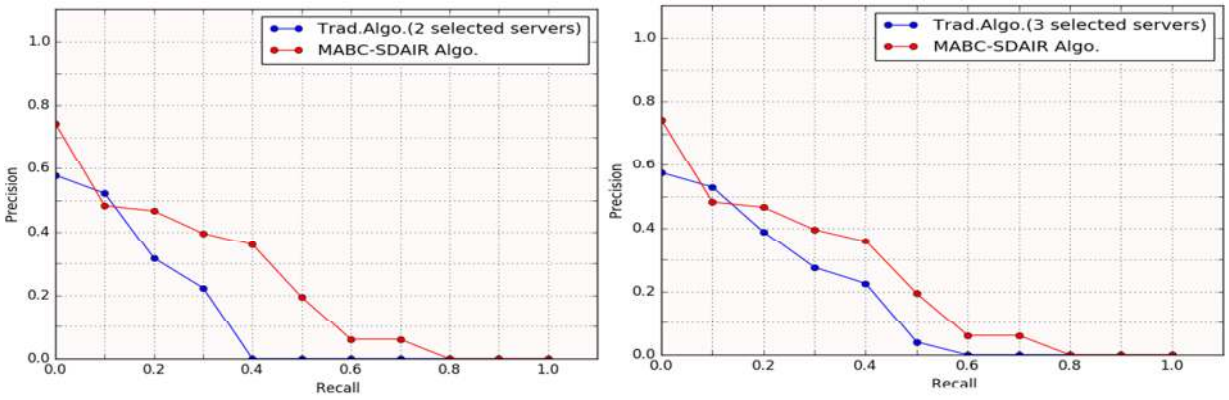

Fig.1. Average Recall-Precision Curves For The First Ten ZAD Queries

Table 2. Average Performance Evaluation Of The NLEL Collection For The First Ten Queries.

Average performance for NLEL collection	Traditional Algo. (2 selected servers)	Traditional Algo. (3 selected servers)	MABC-SDAIR Algo.
Average of documents that are visited for each query out of (11638) documents.	1253	1659	1133
Average of latency (Sec. /Query).	0.521059	0.712359	0.319269
Average of precision.	0.260000	0.300000	0.400000
Average of recall.	0.092932	0.116360	0.144814

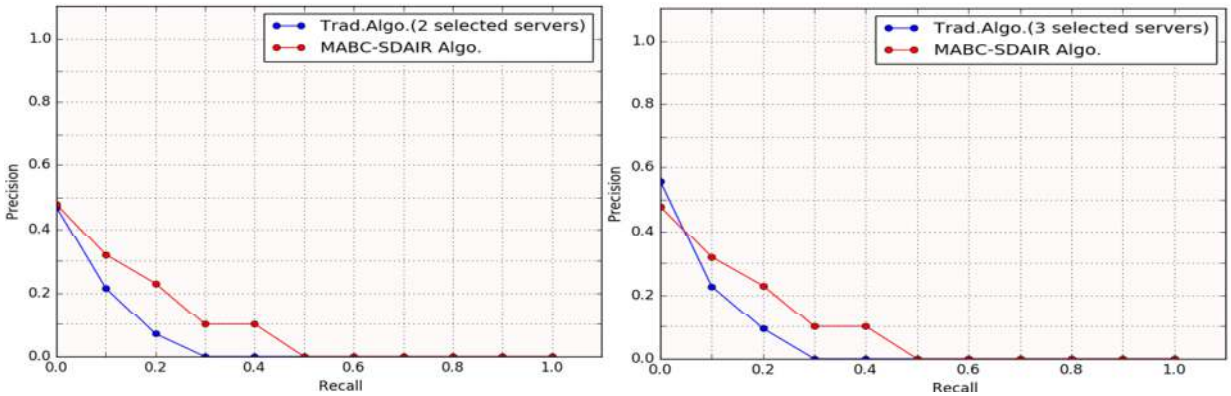

Fig.2. Average Recall-Precision Curves For The First Ten NLEL Queries

To compute the latency, we tend to draw a simple simulation to a client-server database structures. The so called Redis* is a database

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

server that provides a very simple client protocol similar to Telnet [30]. Each Redis server store a database consists of the inverted index and the VSM. If we have four servers, then we have four inverted indexes and four VSMs. The neighbor-docs data structure is stored in a specific server alone.

The consumed times for both traditional and proposed systems are computed depending on the retrieval processes and are recorded as follows:

1. Time spent in the traditional system includes the retrieving of N (sub-collection size) and df (document frequency) those are required to compute the query term weighs, and the retrieving of documents' identifiers from the inverted indexes in the corresponding servers. Also, the time to retrieve the document vectors from the VSMs in the corresponding servers.
2. Time spent in the proposed system includes the retrieving for N and df to compute the query term weighs as in the traditional system, and the time to retrieve the searched document vectors form the VSMs on the corresponding servers and finally the time to retrieve the neighbor-docs from the specific server.

8. DISCUSSION

Although the proposed mechanism for query expansion in MABC-SDAIR algorithm is efficient to find the suitable senses by finding the best synonyms, but it may still be inaccurate from the point of view linguistics. However, it increases the system effectiveness because it depends on the available documents to determine the best synonyms and not on the external resources of the general-purpose dictionaries and thesauruses that may hurt the system performance. On the other hand, the proposed mechanism is very useful to address the problem of lack of the efficient Arabic computational lexical resources.

It is clear that the tested latency in Table 1 and 2 is not strongly influenced by the number of visited documents. The high latency in the traditional systems is due to using the inverted indexes. The hashing search for matching the query terms with the index words in an inverted index and extract the documents' identifiers has a complexity be increased exponentially with the number of existing index words. The random search in the swarm intelligence approaches enables to get results in a polynomial response time. In other words, the proposed system efficiency is directly proportional to the size of inverted index and sizes of inverted lists (posting lists) within inverted index, which is used in traditional systems. In fact, the proposed system will be inefficient if the inverted index is small size and there are a number 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 will be better than the hashing access. Also in large-scale environments, it is expected that the majority of the inverted lists will have larger sizes. Overall, the efficiency results in the two tables prove that the greater the size of the document collection, the greater benefit in response time.

It should also be noted that the result quality maybe not always preserved in the system, this is true due to the nature of the random search. However, the most of the times it achieves results have superiority on traditional systems. This superiority is due to expand the queries and also to reach the documents that are prohibited because the effect of words' weights. In general, the numerical results in Table 1 and 2 exhibit the superiority of the proposed system on the traditional system in terms of the efficiency and the result quality. Fig. 1 and 2 show the superiority of the proposed system in the precision and recall with respect to the preceding appearing of the documents to the user.

9. CONCLUSIONS

In this paper, we developed a method called MABC-SDAIR that uses Artificial Bee Colony (ABC) algorithm with significant modifications for increasing the performance of the Distributed Arabic Information Retrieval (DAIR) systems. The neighbor documents are constructed using the nearest neighbor graph. This is a successful idea to improve the MABC-SDAIR search. The proposed system overcomes the problems that were raised in the traditional DAIR systems. These problems include the high response time due to using the traditional search through the inverted index and also the decline in the results quality due to the resource selection on the one hand as well as the using of the ambiguous words in the query on the other hand. The system tried to find the senses of query words and extract the best synonyms with

* http://redis.io

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the aim to expand the query. The query expansion then increases the results quality while the stochastic optimization search of MABC-SDAIR algorithm increases the efficiency. The proposed system is compared with traditional system which has non-expanded query. ZAD and NLEL are two Arabic corpuses used to test the system performance. The experimental results exhibit the superiority of the proposed system in terms of the precision, recall and latency in comparison to the traditional system.

10. FUTURE WORK

A hybrid bio-inspired meta-heuristic algorithm with a fuzzy logic mechanism can be designed to address the problem of finding the best synonyms weights in the query expansion process in a more interesting performance. Also the using of the stochastic cooperative optimization algorithms is expected to perform best for AIR systems that require the distribution of their tasks.

REFERENCES

[1] Y. Gupta et al., "A new fuzzy logic based ranking function for efficient Information Retrieval system", *Expert Systems with Applications*, Vol.42, No.3, 2015, pp. 1223-1234.

[2] B. Sara and G. Larbi, "Selection of Relevant Servers in Distributed Information Retrieval System", *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, Vol.10, No.5, 2016, pp.724-728.

[3] Y. Rasolofo et al., "Approaches to collection selection and results merging for distributed information retrieval", *In Proceedings of the tenth international conference on Information and knowledge management*, ACM Press, 2001, pp.191-198.

[4] J. Callan, "Distributed information retrieval", *In W. B. Croft, editor, Advances in Information Retrieval.Kluwer Academic Publishers*, (Chapter 5), 2000, pp.127-150.

[5] L.S . Larkey et al., "Light Stemming for Arabic Information Retrieval", *Arabic Computational Morphology Text, Speech and Language Technology*, Vol.38, 2007, pp. 221-243.

[6] M. Saad and W. Ashour, "Arabic Morphological Tools for Text Mining", *In 'EEECS'10 the 6th International Symposium on Electrical and Electronics Engineering and Computer Science' , European University of Lefke, Cyprus*, 2010, pp.112-117 .

[7] T. Rachidi et al., "Arabic user search Query correction and expansion", *In: Proceedings of COPSTIC 2003, Rabat*, 2003, pp. 11-13.

[8] S. Elkateb et al., Arabic WordNet and the Challenges of Arabic, *The Challenge of Arabic for NLP/MT. International conference at the British Computer Society*, 2006, pp. 15-24.

[9] A. Abu-Errub1 et al., "Arabic Roots Extraction Using Morphological Analysis", *IJCSI International Journal of Computer Science Issues*, Vol.11,No.2, 2014, pp. 128-134.

[10] M. Saad, "Mining Documents and Sentiments in Cross-lingual Context", *PhD thesis, Computer Science Dept., Université de Lorraine, France*, 2015.

[11] M.Y. Dahab et al., "A Comparative Study on Arabic Stemmers", *International Journal of Computer Applications*, Vol.125, 2015, pp.38-47.

[12] B. Ghansah and S. Wu, "Survey On Score Normalization : A Case Of Result Merging In Distributed Information Retrieval", *WSEAS Transactions on Information Science and Applications*, Vol.12, 2015, pp. 138-147.

[13] G. Paltoglou, "Algorithms and strategies for sources selection and results merging (collection fusion algorithms) in distributed information retrieval systems", *Phd thesis, Department of Applied Informatics,University of Macedonia, Thessaloniki*, (Chapter 2), 2009, pp. 40-80.

[14] S. Wu and S. McClean, "Result Merging Methods in Distributed Information Retrieval with Overlapping Databases", *Information Retrieval*, Vol.10, No.3, 2007, pp. 297-319.

[15] G. Paltoglou et al., "Results Merging Algorithm Using Multiple Regression Models", *In Proc. 29th European Conference on Information Retrieval*, 2007, pp. 173-184.

[16] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical Report TR06, *Computer Engineering, Department, Erciyes University, Turkey,* 2005.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[17] C. Zhang et al., "An artificial bee colony approach for clustering". Expert Systems with Applications, Vol.37, No.7, 2010, pp. 4761–4767.

[18] G.R. Tankasala, "Artificial Bee Colony Optimization for Economic Load Dispatch o f a Modern Power system", *International Journal of Scientific & Engineering Research*, Vol.3, No.1, 2012, pp. 1-6.

[19] J. Callan et al., "Searching Distributed Collections with Inference Networks", *Proceedings of the ACM-SIGIR 95*, 1995, pp. 21-28.

[20] M. Shokouhi and L. Si, "Federated Search", *Foundations and Trends® in Information Retrieval*, Vol.5, No.1, 2011, pp.1–102.

[21] Z. Saoud and S. Kechid, "Integrating social profile to improve the source selection and the result merging process in distributed information retrieval", *Information Sciences*, Vol.336, No.C , 2016, pp.115-128.

[22] C. Ramya and K.S. Shreedhara, "A Brief Review On The Application Of Swarm Intelligence To Web Information Retrieval", *International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)*, Vol.3, No.1, 2016, pp.60-63.

[23] H. Drias and H. Mosteghanemi, "Bees Swarm Optimization based Approach for Web Information Retrieval", *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2010, 6-13.

[24] A.K. Abdul-Hassan and M.J. Hadi, "Sense-Based Information Retrieval Using Artificial Bee Colony Approach", *International Journal of Applied Engineering Research*, Vol.11, No.15, 2016, pp. 8708-8713.

[25] A.Y. Mahgoub et al., "Semantic Query Expansion for Arabic Information Retrieval", *In: EMNLP: The Arabic Natural Language Processing Workshop, Conference on Empirical Methods in Natural Language Processing, Doha, Qatar*, 2014, pp.87-92.

[26] K. Shaalan et al., "Query expansion based-on similarity of terms for improving Arabic information retrieval", *Intelligent Information Processing VI: Proceedings of 7th IFIP TC12 International Conference, Springer, Heidelberg*, Vol. 385, 2012, pp. 167-176.

[27] H. Khafajeh et al., "Automatic Query Expansion for Arabic Text Retrieval Based on Association and Similarity Thesaurus", *Proceedings of EMCIS*, 2010, pp.1-17.

[28] Y . Benajiba et al., "Implementation of the ArabiQA question answering system's components", *In: Proc. Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium, ICTIS-2007, Fez, Morroco,* 2007, pp. 3–5.

[29] N. Moreau, "Best Practices in Language Resources for Multilingual Information Access", *D5.2 Technical Report: TrebleCLEF project: FP7 IST ICT-1-4-1. TrebleCLEF Project*, (Chapter 3), 2009, pp.7-35.

[30] A. Chinnachamy, "Instant Redis Optimization How-to". Paperback, *Packt Publishing*, (Chapter 1), 2013, 56 pages.