International Journal of Multimedia and Ubiquitous Engineering
Vol.12, No.1 (2017), pp.363-376
http://dx.doi.org/10.14257/ijmue.2017.12.1.31

Sense-Based Information Retrieval Using Fuzzy Logic and Swarm Intelligence

Alia Karim Abdul Hassan[1] and Mustafa Jasim Hadi[2]*

[1]Computer Science Department, University of Technology/Baghdad, Iraq
[2] Computer Science Department, University of Technology/Baghdad, Iraq
[1]hassanalia2000@yahoo.com, [2]mustafa_awadi@yahoo.com

Abstract

Improvement of the quality of information retrieval (IR) using the word sense disambiguation (WSD) was subject of controversy among a lot of authors. However, the recent researches pay tribute to the positive role for using the WSD to improve the IR systems. Query expansion is one of the means to improve the IR using the WSD. Many authors reported that the insertion of synonyms into query after exam the best senses improves the IR quality. However, there are two problems still facing IR systems based on sense, the first is the extent of WSD accuracy, and the second is the delay when addressing a large-scale document collections. This work aims to develop an innovative model to address these two problems. It uses the fuzzy logic to improve the WSD accuracy through tuning the synonyms weights while uses the swarm intelligence, specifically the artificial bee colony (ABC) approach, to address the latency problem. Experimental results show that the proposed model is superior to the traditional model in terms of the precision, recall and latency.

Keywords: *Information Retrieval, Word sense Disambiguation, Query Expansion, Fuzzy Logic, Artificial Bee Colony*

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. It is heavily dependent on Natural Language Processing (NLP), such as the tokenization, stopword removal, and the word stemming. Word Sense Disambiguation (WSD) has become one of the central challenges in the NLP field, it aims to find the correct meaning (sense) of a word in a given context [1]. In the information retrieval, the sense ambiguity can cause the retrieval of irrelevant documents and also the different words which share the same sense can preclude retrieve all the relevant documents. Query expansion looks for the best senses and synonyms of the query words in order to address the word sense ambiguity in IR systems [2]. However, there are two problems raised during design a sense-based IR system: the first is how to avoid the effects of wrong expansion words, and the second is the potential delay when addressing large-scale document collections. The former problem is addressed by many authors using heterogeneous information resources, along with appropriate weighting methods [3]. The latter problem is addressed efficiently using one of the metaheuristic searches [4].

There are usually two types of information resources for disambiguation, the structured and unstructured resources. Although structured resources such as WordNet are very suitable for resolving the word ambiguity, but the using of only the relations defined in these resources for query expansion was reported that they, in many conditions, don't improve performance. The main reason is that structured resources lacked the domain-

[*] Corresponding Author: Mustafa Jasim Hadi . Email: mustafa_awadi@yahoo.com

ISSN: 1975-0080 IJMUE

specific word relations. Therefore, it preferred to be complemented using the unstructured resources such as the corpora and other collection resources. From the other side, it's not preferred to use the unstructured resources alone because some general knowledge may be missing [3]. In this work, the heterogeneity of information resources is addressed using a fuzzy logic mechanism. Fuzzy logic, in general, is a mathematical computation that uses linguistic variables instead of numbers in order to tolerate the imprecision and uncertainty. Fuzzy logic can be a good mechanism for addressing the textual information retrieval because the natural language is used in forming the queries and documents [5]. The fuzzy logic is used in this work to improve the query expansion through estimate the accuracy and importance of synonyms. The other significant problem in IR systems, as we have indicated previously, is the potential delay when addressing large-scale document collections. This problem was addressed by many authors using the metahuristic search. In this work, we used one of the swarm intelligence approaches called Artificial Bee Colony (ABC), but with making significant modifications in order to suit the mechanism of the query expansion. Swarm intelligence approaches, in general, are reported to get a response time with a polynomial rate on a higher computation scale [6-7]. The essential contribution of this work lies in improving IR performance in the term of effectiveness through improving the automatic word sense disambiguation and query expansion, and in the term of efficiency through using a stochastic search rather than the traditional complete search approach.

2. Word Sense Disambiguation and Information Resources

The lexical ambiguity, that appears when a given word has several different meanings, is resolved using what is called in the literature the word sense disambiguation (WSD). The ambiguity in the words is one of the most important open problems in natural language processing (NLP) applications that urged many researchers to innovate various approaches In order to achieve the WSD process. However, there is no WSD approach can guarantee the accuracy, large-scale, and broad-coverage [8].

Information resources for WSD are usually divided into two groups. The first is the structured resources (or the external lexical resources) such as the machine-readable dictionaries, thesauri, and ontologies. The second is the unstructured resources such as the raw corpora and sense-annotated corpora. Depending on the division above, the WSD approaches are also divided into the so-called knowledge-based (or dictionary-based, or knowledge-rich) and corpus-based (or knowledge-poor) approaches. In other words, the knowledge-based approaches use the structured resources such as machine-readable dictionaries, thesauri, and ontologies for disambiguation while the corpus-based approaches use the unstructured resources such as the corpora and the other collocation resources for disambiguation [9]. There are several methods combine both the knowledge-based and the corpus-based approaches into hybrid approaches. The main reason is that the structured resources used in the knowledge-based approaches lack the domain-specific word senses or may be some words do not appear in the available dictionaries or thesauri. For example, with the CACM collection, the authors of the work [10] found that about 9% of the query words do not appear in the given Longman Dictionary of Contemporary English (LDOCE) at all, and that another 22% are used either in a domain-specific sense or in a sense called "marginal" (such as file, language, pattern, and code, *etc*.) that violates the semantic restrictions, or it is used in a sense that is somewhat different from the one listed in the dictionary [10].

The corpus-based approaches are able to utilize the domain specific information, since they use the corpus itself in an automatic manner to discriminate the ambiguity of the words within a set of documents relating to a certain domain. However, it's not preferred to use only the corpus information because some general knowledge may be missing. Also, it is difficult to capture the relations between certain words that share the same sense. For example, ‘‘tumor’’ and ‘‘tumour’’ denote the same sense, but would never (or at least not frequently enough) appear in the same document. This is because it is not expected that an author puts a variety of words have the same sense in the same document. This kind of relationship can be found in the structured resources that are general-purpose lexicons. Since there are different advantages and disadvantages each type of resource, combining them provides a valuable tool for improving the information retrieval [3]. The sense-based information retrieval systems aim to determine and using the suitable senses of the words that have multiple senses within their databases and/or queries to improve the retrieval quality. If the system is only interested in resolving the ambiguity in the query words, then the synonyms of those suitable meanings are used for the query expansion purpose.

The most common structured resource for resolving the ambiguity of words is the WordNet. WordNet has been developed at Princeton University and is a lexical database system used online. This system models the lexical knowledge of English native speaker. It consists of nearly 100,000 terms classified hierarchically. Noun, verb, adjective and adverb for each word are grouped into synonym sets. The synonym sets of all words are also organized into senses. Synonym sets can also be related to Hyponym/ Hypernym, and the Meronym/Holonym [11]. In this work, for the purpose of query expansion, we combine both the structured and unstructured resources in a single system. WordNet as a structured resource and CACM and NPL corpora (document collections) as unstructured resources are used for testing the experiments.

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 [12]. Many IR models are mentioned in the literature. Vector space model (VSM) is the most widely used in which documents and queries are stored in weights vectors. The weight vector of a document d is defined as $\vec{d} = (w_{d,1}, w_{d,2}, \ldots, w_{d,n})$ while the weight 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 is calculated 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 above vectors as follows:

$$cos\ (q,d) = \frac{\vec{q}.\vec{d}}{|\vec{q}|.|\vec{d}|} = \frac{\sum_{t=1}^{n} w_{q,t} w_{d,t}}{\sqrt{\sum_{t=1}^{n} w_{q,t}^{2}}.\sqrt{\sum_{t=1}^{n} w_{d,t}^{2}}} \qquad (3.1)$$

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 [12].

The most common traditional search approach in IR applications is the search based on the inverted index. Inverted index is a collection of lists, one per term, recording the identifiers of the documents containing that term and also the term frequency in each document [13]. Through using the inverted index, the search complexity of query-document with non-zero similarity is reduced at a phenomenal rate. However, the inverted index file may become untreatable for an environment with huge documents. Metaheuristic approaches can get a response time with a polynomial rate on a higher computation scale [6-7].

4. Fuzzy Logic in Information Retrieval

Fuzzy logic is a convenient formal modeling tool in the area of information retrieval (IR). The retrieving of a relevant document in respect of a query or the statistical importance of a term within a document/query opened the way towards the fuzzy logic based interpretation. Although there are many interesting approaches on the application of fuzzy logic in IR, but nevertheless there is no powerful and comprehensive approach based on fuzzy logic in dealing with all aspects of imperfect information in IR. The first step in the direction of applying fuzzy logic in IR is to convert the traditional retrieving using the binary logic model to the retrieving using the multivalued logic model. The document is treated as a fuzzy set of terms and the membership of a term reflects its importance in that document. The next step is to allow for weights to be associated with terms in the query [14]. Basically, there are three semantics of weights in queries are defined as follows [15]:

1. Importance semantic: the weight of a query term is defined as a measure of the relative importance of this query term in respect of other query terms.
2. Threshold semantic: the weight of a query term is defined as a satisfaction requirement for this query term to be considered during the matching process.
3. Perfection semantic: the weight of a query term is defined as a description of the perfect documents, which contain it, desired by the user.

The representation of query terms in context of the importance has given a widely recommendation by the authors. Usually, the interval [0,1] is the range of the importance degree. However, it is not realistic to expect that the expression of a number will be precise. Thus, the importance is preferred to be expressed in fuzzy linguistic terms as “important”, “very important”, *etc*. [14]. Fuzzy logic systems, in general, use fuzzy logic for formulating the mapping from a given input to an output. This requires three basic stages are the fuzzification, rule decision making, and defuzzification [16]. Fuzzification is a process to transform the crisp values into a fuzzy linguistic range using a membership function. Membership function shape can be either triangular, trapezoid, singletone or bell-shape. If we interested with the processing time, the triangular with 50% overlap between the adjacent membership functions is more preferred since it contributes to a less computational process time [17]. On the rule decision making stage, fuzzy rules are represented in the general form as “IF A THEN B” where A and B are propositions containing linguistic variables. The consequent B is reshaped using a function associated with the antecedent A. The consequent of a rule is a fuzzy set represented by a membership function. The rules must be combined in some manner in order to make a decision [16]. Defuzzification is the last stage where the fuzzy form is transformed to crisp output. There are several possible methods for defuzzification, two methods usually used are the centroid method and mean-maximum method [17]. In this work, the triangular membership function is used in the fuzzification process while the centroid method is used in the defuzzification process. The purpose of the use of fuzzy logic in this work is to improve the weights of synonyms through taking the outputs from the two approaches, knowledge-based and corpus-based approaches, as the inputs of the fuzzy system. The output of the fuzzy logic system contributes as a supporting factor in formulating the synonyms weights equation.

5. Artificial Bee Colony Algorithm

Artificial Bee Colony (ABC) algorithm is a stochastic-based metaheuristic algorithm falls under the umbrella of swarm intelligence. The ABC algorithm is introduced by Karaboga [18]. The main steps of the ABC algorithm are described as follows [19].

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:

$$x_{ij} = x_{\min j} + \text{rand}\ [0, 1].\ (x_{\max j} - x_{\min j}) \tag{5.1}$$

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. The 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:

$$v_{ij} = x_{ij} + \phi_{ij}\ .(x_{ij} - x_{kj}) \tag{5.2}$$

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

$$p_i = a \times \frac{f_i}{f_{\max}} + b \tag{5.3}$$

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. (5.1) [20].

6. Related Works

In the work [21], the authors presented in their paper a brief review on the application of swarm intelligence to information retrieval with focused on the large-scale databases and the Web. Although the mentioned works improve response time and the results quality, but with that they were based only on keywords to compute the similarity between queries and documents and lack to consider the sense of the word in its context. Finding out the correct sense of a word in a text for improving the information retrieval is still a challenge factor depending on the views of many authors. The authors on the work [1] presented a literature review about the combination between word sense disambiguation (WSD) and information retrieval (IR). They list some of researches and studies about the using of WSD in IR. Some of the presented works in the list reveal that there is a controversy is evident in the possibility of using the WSD with IR. Among the most prominent of these works is for the author in [22], who presented a review about the controversy in the ability of the WSD to improve IR and noted the authors who answered by yes or no for the question “Does WSD helps in IR?”. The our previous work in [4] 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 IR systems. The work supported the direction who said that the WSD can improve the IR. The ambiguity was resolved by using the Simplified Lesk algorithm. CACM and NPL corpuses were used for conducting the experiments. However, for the dynamic databases, the offline complete disambiguation of the whole document collection is still complex and inefficient, especially for the large scale corpora. The successful alternative is only resolving the ambiguity and expanding the query words without the need for resolving the ambiguity of the words in the whole corpus. More works are related to improving the IR using the query expansion. In the query expansion mechanisms, the new terms are added into the original query either from the existing corpus (*i.e*. the unstructured resource) or from an external lexicon resource (*i.e*. the structured resource). The earliest work in [23] tried to extract the new terms from processed documents using the relevance feedback and adding them into the query for improving the system performance. In other trends, the work in [24] used the WordNet as an external thesaurus to disambiguate the word senses for the text retrieval [25] while the work in [26] used different types of thesaurus for query expansion and, in the same context, to avoid the wrong term expansion, the authors presented a new weighting term method. This weighting term method is designed so that the weight of expansion terms depends on similarity measures in all types of thesauri in addition to the dependence on all query terms. Their experiments exhibited that the use of the heterogeneous thesauri gives better retrieval results than just using one type of thesaurus. The work in [27] used the WordNet and the web search to determine the sense of expansion terms, the query expansion was carried using pseudo feedback. More recently the work in [28] introduced a new method of query expansion based on relevance feedback and latent semantic analysis. This method finds the relative terms to the topics of user original query based on relevant documents selected by the user in relevant feedback step. The results proved that the method provides a better representation of the user query and increases his/her satisfaction.

7. Proposed Sense-Based IR System

Research into the automatic ambiguity solution for the senses of words has been tackled by a lot of authors. However, several factors are still under consideration, for example, if we expand a query using some wrong words from a thesaurus, there is a high probability that the output will not be satisfactory. Also, the more resources used for resolving ambiguity or the increased complexity of the disambiguation algorithms produce a system owns a slow response time. This work aims to find an efficient and effective method to disambiguate the query words, extract their synonyms, and tuning of the synonyms weights. The work depends on two soft computing techniques inspired from the fuzzy logic and the swarm intelligence. As we mentioned in an advance section that combining of the knowledge-based and the corpus-based approaches for solving the ambiguity was reported in many works to introduce better results from the using of only a one approach. However, unlike previous work, this work is not concerned to select the senses or the synonyms depending on the heterogeneity of the two approaches, but the focus here is to support the strength of selection of the knowledge-based approach by utilizing of the corpus-based approach. This supporting is represented by balancing the synonyms weights through the assumption that if the weight of a synonym produced by the knowledge-based approach is higher than the weight of another synonym produced by the corpus-based approach, then the synonym produced by the knowledge-based approach is more important and we expect that a small increasing to the original weight can get more relevant than the original one. In the same manner, if the weight of a synonym produced by the corpus-based approach is higher, then the synonym produced by the knowledge-based approach is lower important and we expect that a small decreasing to the original weight can get better results. This inverse relationship of the two approaches is represented in this work using the fuzzy logic. Of course, if the two approaches find the same synonym, then the weight remain unchanged. In knowledge-based approach, the definitions of the senses for the query words are extracted using the WordNet. The definition of a sense of word which has a maximum similarity with the word context reflects the best sense of that word. If we determined the best sense, we select at random a synonym from the list of synonyms related to that sense. To decrease the matching and searching time, we store all the words in all the definitions in an inverted index and dealing with the word context as a query directed to the inverted index. For distinguishing this inverted index from the original inverted index that is used in the information retrieval, we call this inverted index as "special inverted index". To get a more accurate response, we associate each word in the definition with a weight represented by *tf-idf* weighting scheme. Also, for distinguishing this *tf-idf* weight from the original *tf-idf* weight used in the information retrieval, we call this *tf-idf* weight as "special *tf-idf* weight". The definitions are retrieved and ranked using the cosine similarity measure and only the first definition in the ranked list is considered as the winner. Since the winner definition is associated with a sense, then that sense is the winner among the others and the related set of synonyms in turn is the winner set. The main objective of the knowledge-based approach is extracting the best synonym and associates it with a weight calculated by *tf-idf* weighting scheme. The proposed sense-based IR system consists of four algorithms called Knowledge-Based Algorithm, MABC Algorithm, Tuning Weights Algorithm, and Expand Query & Matching Algorithm respectively. Also, we called all these algorithms as the FMABC algorithm, where F refers to the Fuzzy logic while MABC refers to the modified ABC. The first algorithm, Knowledge-Based Algorithm, is illustrated below:

Algorithm 1: Knowledge-Based Algorithm

Input:

$Q = \{w_{q,i} | i = 1..n\}$ /* $w_{q,i}$ is a word (associated with a *tf-idf* weight) in a query Q*/

$w_{q,i}^{G}$= $\{w_{q,i}^{g} | g = 1..k\}$, $w_{q,i}^{g}$=$\{w_{q,i}^{g,j} | j = 1..m\}$ /* $w_{q,i}^{g,j}$ is a word (associated with *a special tf-idf weight*) in the gloss $w_{q,i}^{g}$. The glosses are extracted from the WordNet*/

Output:

S^{K}=$\{w_{q,i}^{s^k} | i = 1..n\}$ /*$w_{q,i}^{s^k}$ is the best synonym (associated with its *tf-idf* weight) of a word $w_{q,i}$*/

Extracting Best Synonyms:

```
For i = 1 to $n$ do
    Get $w_{q,i}^{G}$ from the matching process using the special inverted index.
    Calculate the cosine similarity as follows:
        For g=1 to $k$ do
            $cos\,(Q, w_{q,i}^{g}) = \frac{Q.w_{q,i}^{g}}{|Q|.|w_{q,i}^{g}|}$     /* Eq. (3.1) */
        End for
    Rank the cosine similarities in decreasing order. Consider the first is the winner
    definition. /*The winner definition refers to the winner sense*/
    Repeat
        Select at random one synonym from the set of synonyms that is associated with
        the wining sense /*Using the WordNet*/
        If the synonym is found in the original inverted index (not the special) then
            Calculate its $tf-idf$ weight.
            Break
        Else: try another synonym
    Until there is no remaining synonym in the set
End for
```

After we get a set of the best synonyms with their weights for all the query words using Algorithm 1, we now try to find another set of synonyms depends on the corpus-based approach. Before we begin with the proposed corpus-based approach, we first associate each query word with its set of senses using the WordNet and in turn select at random one synonym for each sense. Now we intend to find the best synonyms (that correspond to the best senses) through using the proposed corpus-based approach. The proposed approach is designed using a swarm intelligence approach called Artificial Bee Colony (ABC) approach. The ABC approach is modified to suit the required task. The Modified ABC (MABC) is intended for two search processes, the first is to search the most relevant documents and the second is to search the best synonyms using the corpus as a resource. The MABC algorithm is illustrated in Algorithm2 as follows:

Algorithm 2: MABC Algorithm

Input:

$D_{n\times m}$ /* Set of documents represented by a vector space model */

$G^{s} = (V, E, S)$ /* document similarity graph*/

$Q = \{w_{q,i} | i = 1..n\}$ /* $w_{q,i}$ is a word (associated with a *tf-idf* weight) in a query Q*/

$S^{C,list}$=$\{w_{q,i}^{s^{c,list}}$ | $i = 1..n\}$ /*each $w_{q,i}$ is associated with a list of different synonyms, one synonym for each sense, their initial fitnesses equal zero. The synonyms are extracted from the WordNet*/

Output:

doc^{global}/*best candidate relevant document*/

S^{C}=$\{w_{q,i}^{s^{C}}$ | $i = 1..n\}$ /*$w_{q,i}^{s^{C}}$ is the best synonym (associated with its *tf-idf* weight) of a word $w_{q,i}$*/

Search Mechanism:

```
Begin
Initialize the population: Set of random food sources (documents) selected from $D_{n\times m}$.
Calculate the fitness values of food sources in the population. Update the synonyms
fitnesses of query words.
cycle = 1
Repeat
  – Memorize $doc^{global}$ and the best fitness so far for each synonym.
  – Determine the neighbors $ni\_docs_{i}^{local}$ from $G^{s}$of the chosen food sources $doc_{i}^{local}$
    for the employed bees and evaluate them. Update the synonyms fitnesses of query
    words.
  – Determine the neighbors $ni\_docs^{global}$ from $G^{s}$of the chosen food sources $doc_{i}^{local}$
    based on the probability for the onlooker bees and evaluate them. Update the
    synonyms fitnesses of query words.
  – Produce the new food sources for the abandoned food sources using the scouts after
    scattering them throughout $D_{n\times m}$. Calculate their fitness values. Update the
    synonyms fitnesses of query words.
  – cycle = cycle + 1
Until cycle = $I_{max}$
Calculate $tf$-$idf$ weight for best synonyms.
End
```

The successful idea to improve the search in the above algorithm is the document similarity graph $G^{\varepsilon} = (V, E, S)$. The neighbor documents (named for short as ni-docs) are constructed offline using the ε −neighborhood document similarity graph G^{ε} after the vector space model $D_{n\times m}$ for the document collection has been completed. Let $G = (V, E, S)$ 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 S 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 s_{ij} equal to the cosine similarity. The ε −neighborhood document similarity graph G^{ε} is the document similarity graph $G^{\varepsilon} = (V, E, S)$ in which each edge e_{ij} in the graph can connect two nodes (documents) with weight s_{ij} represents the cosine similarity result for the two nodes d_i and d_j if and only if the cosine similarity result $\geq \varepsilon$.

Initially the documents are randomly chosen in an integer interval from 1 to the collection size. The fitness function in the algorithm is the cosine similarity between a query and a document. At each time the fitness is calculated, a synonym weight is updated depending on the fitness and the best synonym is memorized. The updating of the weights depends on the assumption that each document can be considered as a gloss or a definition of that synonym and the query is the context where we need to find the suitable word meaning (or the correct 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. 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 fitness is larger than the old one. At each iteration, the new documents are evaluated and the synonyms weights are updated. Local Neighbor-docs (named for short as ni_docs^{local}) refers to the documents that are nearest to the current document (named for short as doc^{local}) within the population, while global Neighbor-docs (named for short as ni_docs^{global}) refers to the documents that are nearest to best document that is known so far (named for short as doc^{global}) 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 near to the current document doc^{local}, *i.e.* from ni_docs^{local}. In the onlooker phase, the neighbor's document is replaced with a document selected depending on Eq. (5.3). 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 doc^{global}, *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 collection size. After the iterations reach to the desired limit, we gain the best document at all doc^{global} and a best synonym for each query word.

After we get two groups of synonyms (one synonym for each sense) with their weights from using algorithms 1 and 2, *i.e.* S^K={$w_{q,i}^{s^k}$ | $i = 1..n$} and S^C={$w_{q,i}^{s^C}$ | $i = 1..n$}, now we aim to recomputed the weights of the synonyms produced from knowledge-based approach, S^K, by looking at the weights of the synonyms produced from the corpus–based approach, S^C. The weights of synonyms are then tuning using a fuzzy logic system described in Algorithm3 below:

Algorithm 3: Tuning Weights Algorithm
Input:
S^K={$w_{q,i}^{s^k}$ | $i = 1..n$},S^C={$w_{q,i}^{s^C}$ | $i = 1..n$}.

```
Output:
$S^F$={$w_{q,i}^{sf}$ | $i = 1..n$} /*$w_{q,i}^{sf}$ is a synonym (associated with its tuned weight) of a word
$w_{q,i}$*/
1. Extract Fuzzy support:
Begin
Antecedents (Inputs):
    Universes: How important was the weight of a synonym in $S^K$and another in $S^C$, on a
               scale of 0 to 1?
    Fuzzy set: high, medium, low importance.
Consequents (Outputs):
    Universe: How much should we support the weight of a synonym in $S^K$, on a scale of
              0 to 1?.
    Fuzzy set: low, medium, high support.
Fuzzy rules:
    R1: IF (high-important-knowledge) AND NOT (high-important-corpus) THEN high-
        support.
    R2: IF (high-important-knowledge) AND (high-important-corpus) THEN medium-
        support.
    R3:IF (medium-important-knowledge) AND (low-important-corpus) THEN medium-
        support.
    R4: IF (medium-important-knowledge) AND NOT (low-important-corpus) THEN
        low-support.
    R5: IF (low-important-knowledge) THEN low-support
End.
2. Tuning Weights Process
Begin
    Re-weight of the synonyms in $S^K$ depending on the fuzzy support using the following
    equation Eq.7.1:
    $Synonym\_weight_{new}$ =0.8* ($Synonym\_weight_{old}$)+ 0.2*(fuzzy support)        (7.1)
End.
```

For example, Let us take an example to use the fuzzy logic for the queries of the CACM collection, for the query 5:

> *"I'd like papers on design and implementation of editing interfaces, window-managers, command interpreters, etc. The essential issues are human interface design, with views on improvements to user efficiency, effectiveness and satisfaction".*

The word "design" has several senses each one has several synonyms. Through using the WordNet, we can extract all the senses and then all the synonyms for this word "*design*". However, not all synonyms must be taken into consideration, we first must exclude the synonyms that are morphologically similar to the original query words in order to avoid the repetition. For example, we have a synonym "*plan*" in the sense2, "*blueprint*" and "*pattern*" in the sense3, "*figure*" in the sense4, "*purpose*", "*intent*", "*intention*", and "*aim*" in the sense5, "*innovation*", "*excogitation*", and "*conception*" in the sense8, "*project*" and "*contrive*" in the sense9. Other senses such as 1,6,7,10,11,12 are ignored because they have only a synonym that similar to "*design*" or have other synonyms are repeated. After applying Algorithm1, we get the synonym "*plan*" among all the synonyms in all the given senses and after applying Algorithm2, we get the synonym "*pattern*" among all the synonyms in all the given senses. The two inputs "*plan*" with a *tf-*

idf weight equals (0.18) and "pattern" with a *tf-idf* weight equals (0.15) are entered into the fuzzy system in Algorithm3. The output of the fuzzy system (fuzzy support) equals (0.35) is extracted through aggregated the memberships and the finding of the centroid result as it is illustrated in the Figure (1) below:

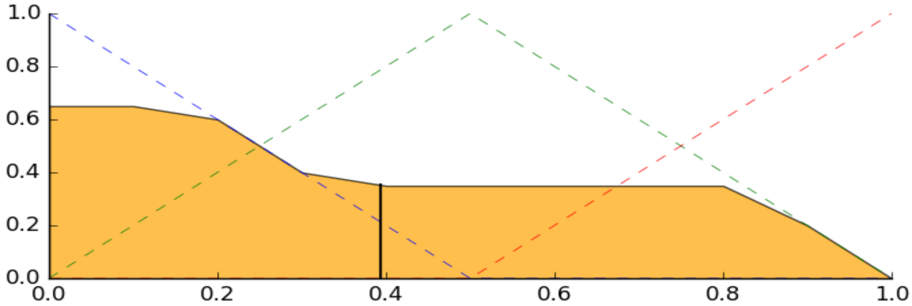

Figure 1. Aggregated Membership and Centroid Result (Line)

The fuzzy support has an important role to understand the importance of a synonym selected form knowledge-based approach. In this example, the synonym "*plan*" has gotten a support 35% for changing its original weight but within a range determined depending on a proposed equation, Eq.(7.1). Equation (7.1) contributes to avoid an imbalance that can happen between the synonyms weights and the other weights of original query words. Now we get a unique set of synonyms with suitable and fair weights. These synonyms become ready to be added to the original query to produce the expanded query. The final task is to match the expanded query with the best candidate relevant document doc^{global} (that is gained from Algorithm2) and its ni_docs^{global}. This is illustrated in Algorithm4 as follows:

Algorithm 4: Expand Query & Matching Algorithm
Input:
$Q = \{w_{q,i} | i = 1..n\}$, doc^{global}, ni_docs^{global}, S^F=$\{w^{sf}_{q,i} \mid i = 1..n\}$.
Output:
Ranked list consist of the best relevant documents organized in descending order.
Expanding and Matching Process:

```
Begin
  Add the synonyms in $S^F$to Q, name the result as $Q^{expanded}$.
  Add $doc^{global}$ to its $ni\_docs^{global}$ list, name the result as $docs^{candidate}$.
  Calculate the cosine similarity between $Q^{expanded}$ and $docs^{candidate}$.
  Sort the results in descending order.
End.
```

8. Experimental Results

The proposed system is experimented on two different corpuses CACM (3204 documents, 64 queries with their relevance judgments) and NPL (11429 documents, 93 queries with their relevance judgments). These corpuses are well-known and used in many research works for evaluating IR systems. The proposed system is run on a personal computer (Core-i5 @2.50 GHz, RAM 6GB, 64-bit operating system). The experimental evaluations focus on the comparison between the proposed system and the traditional inverted-index system that does not have an expanded query. To highlight the comparison, a sample of the first ten queries is selected from each collection to evaluate the results. Tables (1) and (2) show the performance average within the rank 10 with respect to the first ten queries of CACM and NPL collections respectively. Figures (2) and (3) 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.

Table 1. Performance Evaluation for First Ten CACM Queries

Average performance for CACM collection	Traditional Algo.	FMABC Algo.
Average of documents that are visited for each query out of (3204) documents.	1074	796
Average of latency (Sec. /Query).	0.184963	0.158920
Average of precision.	0.230000	0.300000
Average of recall.	0.270198	0.339841

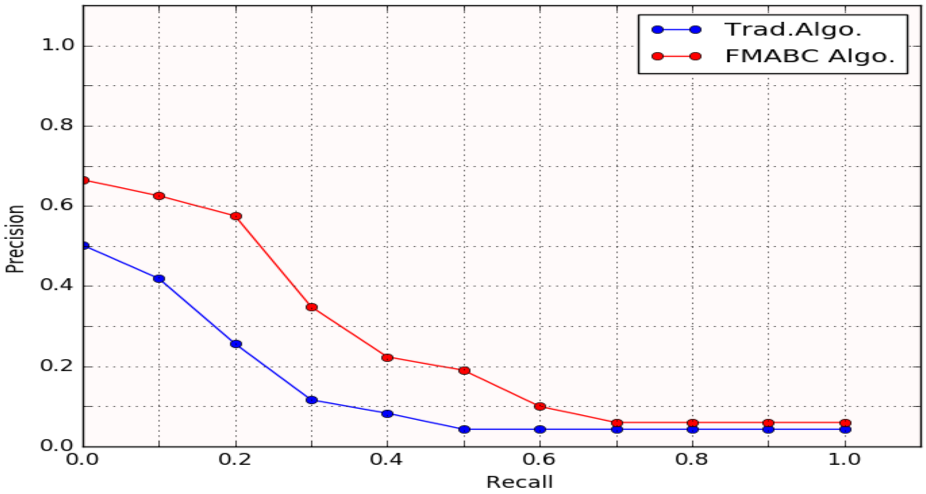

Figure 2. Average Recall-Precision Curves for First Ten CACM Queries

Table 2. Performance Evaluation for First Ten NPL Queries

Average performance for NPL collection	Traditional Algo.	FMABC Algo.
Average of documents that are visited for each query out of (11429) documents.	2327	1561
Average of latency (Sec. /Query).	1.093553	0.709708
Average of precision.	0.200000	0.230000
Average of recall.	0.269174	0.299164

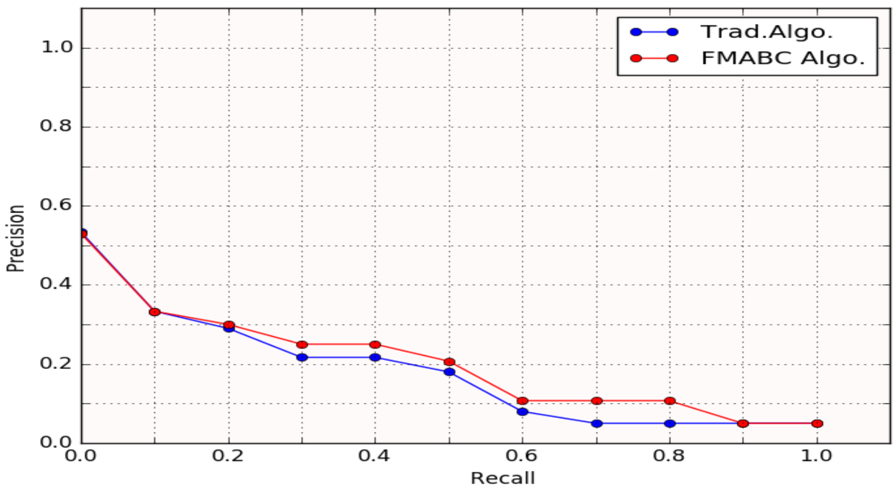

Figure 3. Average Recall-Precision Curves for First Ten NPL Queries

The average point of the11-points in Figure (2) equals (0.149) for traditional system and equals (0.270) for the proposed system while in Figure (3) it equals (0.186) for traditional system and equals (0.208) for the proposed system.

9. Conclusions

In this paper, we developed a model uses the fuzzy logic and swarm intelligence for increasing the performance of information retrieval systems. A swarm intelligence approach called artificial bee colony (ABC) is modified to improve the searching process using a nearest neighbor graph. The proposed system has overcome on two problems. The first is the time consuming problem that is raised in the systems that use traditional search through the inverted index. The second problem is the drop in the quality due to the ambiguous words in the query. The proposed system tried to find the best senses of query words with extract the best synonyms and tuning their weights using the fuzzy logic. This was done depending on both the lexicon and the content of the document collection itself. The expanded query that is produced after adding the best synonyms contributes to increase the results quality while the stochastic optimization search of the modified ABC algorithm increases the efficiency. The experimental results exhibit the superiority of the proposed system in terms of the precision, recall and latency in comparison to the traditional system.

References

[1] B. F. Z. Al_Bayaty and S. Josh, "Word Sense Disambiguation (WSD) and Information Retrieval (IR): Literature Review", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 2, **(2014)**, pp. 722-726.

[2] F. B. D Paskalis, M. Khodra, F. B. D. Paskalis and M. L. Khodra, "Word Sense Disambiguation In Information Retrieval Using Query Expansion", International Conference on Electrical Engineering and Informatics (ICEEI), **(2011)**, pp. 1-6.

[3] R. Mandala, T. Tokunaga, and H. Tanaka "Query expansion using heterogeneous thesauri", Information Processing and Management, vol. 36, **(2000)**, pp. 361-378.

[4] A. K. A. 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

[5] L. N. A. Al-Aziz, "Enhancement of information retrieval ranking using fuzzy logic", The British University in Dubai, UAE, Master thesis, **(2011)**.

[6] 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, vol. 1, **(2010)**, pp. 6-13.

[7] H. Drias, “Parallel Swarm Optimization for Web Information Retrieval”, IEEE Third World Congress on Nature and Biologically Inspired Computing, **(2011)**, pp. 249-254.

[8] A. Montoyo, A. Su´arez, G. Rigau, and M. Palomar, “Combining Knowledge-and Corpus-based Word-Sense-Disambiguation Methods”, Journal of Artificial Intelligence Research, vol. 23, **(2005)**, pp. 299-330.

[9] R. Navigli, “Word Sense Disambiguation: A Survey”, ACM Computing Surveys, vol. 41, no. 2, **(2009)**, pp. 1-69.

[10] R. Krovetz and W. B. Croft, “Lexical Ambiguity and Information Retrieval”, ACM Transactions on Information Systems, vol. 10, no. 2, **(1992)**, pp. 115-141.

[11] G. Varelas, E. Voutsakis, E. G. M. Petrakis, E. E. Milios and P. Raftopoulou, “Semantic Similarity Methods in WordNet and their Application to Information Retrieval on the Web”, 7th ACM Intern. Workshop on Web Information and Data Management, **(2005)**, pp. 10-16.

[12] R. B. Yates and B. R. Neto, “Modern Information Retrieval”, Addison Wesley Longman Publishing Co. Inc.(ACM), 1st ed., (Chapter 1), **(1999)**, pp. 1-17.

[13] J. Zobel and A. Moffat, “Inverted Files for Text Search Engines”, Journal ACM Computing Surveys (CSUR), vol. 38, no. 2, **(2006)**, pp. 1-56.

[14] S. Zadrozny and K. Nowacka, “Fuzzy information retrieval model revisited”, Fuzzy Sets and Systems, vol. 160, **(2009)**, pp. 2173–2191.

[15] E. H. Viedma, “Modeling the Retrieval Process for an Information Retrieval System Using an Ordinal Fuzzy Linguistic Approach”, Journal of the American Society for Information Science and Technology, vol. 52, no. 6, **(2001)**, pp. 460-475.

[16] S. Manna and B. S. U. Mendis, “Fuzzy Word Similarity: A Semantic Approach Using WordNet. Fuzzy Systems (FUZZ)”, 2010 IEEE International Conference, **(2010)**, pp.1-8.

[17] S. M. Ayob, Z. Salam and N. A. Azli, “Simple P1 Fuzzy Logic Controller Applied in DC-AC Converter”, First International Power and Energy Conference PEC, Putrajaya, Malaysia, **(2006)**, pp. 28-29.

[18] D. Karaboga, “An idea based on honey bee swarm for numerical optimization”, Technical Report TR06, Computer Engineering, Department, Erciyes University, Turkey, **(2005)**.

[19] C. Zhang, D. Ouyang and J. Ning, “An artificial bee colony approach for clustering”, Expert Systems with Applications, vol. 37, no. 7, **(2010)**, pp. 4761-4767.

[20] 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.

[21] 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.

[22] H. T. Ng, “Does Word Sense Disambiguation Improve Information Retrieval?”, ESAIR’11, Glasgow, Scotland, UK. ACM, **(2011)**, pp. 17-18.

[23] D. Harman, “Relevance Feedback Revisited”, in Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, **(1992)**, pp. 1-10.

[24] E. M. Voorhees, “Using WordNet to Disambiguate Word Senses for Text Retrieval”, in Proceedings of the 16th ACM-SIGIR Conference, **(1993)**, pp. 171-180.

[25] F. B. D. Paskalis and M. L. Khodra, “Word Sense Disambiguation In Information Retrieval Using Query Expansion”, 2011 International Conference on Electrical Engineering and Informatics, **(2011)**, pp. 1-6.

[26] R. Mandala, T. Tokunaga and H. Tanaka, “Query expansion using heterogeneous thesauri”, Information Processing and Management, vol. 36, **(2000)**, pp. 361-378.

[27] S. Liu, C. Yu, and W. Meng, “Word Sense Disambiguation in Queries”, in CIKM'05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, **(2005)**, pp. 525-532.

[28] M. Rahimi and M. Zahedi, “Query expansion based on relevance feedback and latent semantic analysis”, Journal of AI and Data Mining, vol. 2, no. 1, **(2014)**, pp. 79-84.