Sense-Based Arabic Information Retrieval Using Harmony Search Algorithm

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
[2] mustafa_awadi@yahoo.com

Abstract: ***Information Retrieval (IR) is a field of computer science that deals with storing, searching, and retrieving documents that satisfy the user need. The modern standard Arabic language is rich in multiple meanings (senses) for many words and this is substantially due to lack of diacritical marks. The task for finding appropriate meanings is a key demand in most of the Arabic IR applications. Actually, the successful system should not be interested only in the retrieval quality and oblivious to the system efficiency. Thus, this paper contributes to improve the system effectiveness by finding appropriate stemming methodology, word sense disambiguation, and query expansion for addressing the retrieval quality of AIR. Also, it contributes to improve the system efficiency through using a powerful metaheuristic search called Harmony Search (HS) algorithm inspired from the musical improvisation processes. The performance of the proposed system outperforms the one in the traditional system in a rate of 19.5% while reduces the latency in an approximate rate of 0.077 second for each query.***

Keywords: Arabic Information Retrieval, Word sense Disambiguation, Query Expansion, Harmony Search.

1. Introduction

Information retrieval (IR) is a field of computer science that deals with storing, searching, and retrieving documents that can be texts, Web pages, images, and videos. The textual IR is heavily dependent on the natural language processing (NLP), such as the tokenization, stopword removal, and the word stemming. Most of IR researches are concerned to manipulate the documents which are written in the English language. In contrast, there are a few IR researches are concerned with Arabic language. This due to the lack of publicly freely accessible Arabic corpuses [1] on the one hand, as well as the morphological and semantic complexity of the Arabic language on the other hand. The traditional Arabic orthography has diacritics (i.e. adding vowel sounds of the Arabic words), but at present, most Arabic documents are written in undiacritized format that is called "modern standard Arabic". The lack of diacritics causes a lot of morphological and semantic ambiguity in Arabic language. Arabic language dissimilar to English language, it is a highly inflectional and derivational language and it has no capital letters precede words to contribute to reveal named entities [2]. Word Sense Disambiguation (WSD) has become one of the central challenges in NLP field, it aims to find the correct meaning (sense) of a word in a given context [3]. In Arabic, there are a lot of words can be derived from a single word root. Thus, Arabic IR (AIR) 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 [4]. This mechanism requires finding out equivalent word alternatives (synonyms) for all or some of query words after applying an appropriate disambiguation technique.

In this work, the proposed AIR system is tested on Arabic corpus called Zad-Al-Ma'ad, namely ZAD in short, has 2730 Arabic documents, 25 Arabic queries, and supported by relevance judgments. It is written by the Islamic scholar "Ibn Al-Qyyim" [5],[6]. We annotate all the words in ZAD corpus, offline for the documents and online for the queries, by exploiting similarities identified among the word senses using the path-based similarity method. The Arabic WordNet (AWN) is the thesaurus used for extracting the senses. AWN is a free lexical resource for Arabic language based on the well-known Princeton WordNet (PWN) for English [7]. The work is compared with the traditional baseline search. The traditional baseline search is the inverted index-based search that depending on the matching between the non-semantic relation words, keywords, of both the query and the documents thorough the inverted index. Before the search process, all the words/keywords in the query and inverted index are stemmed in a preprocessing stage with the exclusion of the diacritics, stopwords, punctuation marks, and special characters. The words also are weighted by tf-idf score and stored using a common retrieval model is the vector space model (VSM). Generally, the traditional baseline search is

used by a lot of authors for comparing results and can be created by the author her/himself or using existing text retrieval toolkits such as Lucene[1], Lemur[2] and Terrier[3].

The essential contribution of this paper lies in improving AIR performance in terms of effectiveness through including the automatic word sense disambiguation and query expansion and in terms of efficiency through using a stochastic search rather than the traditional complete search approach.

2. Arabic Morphological and Semantic Analysis

2.1 Arabic Lexical Stemmers

Arabic language has about 10,000 roots each one can generate hundreds of lexical forms of different meanings [8]. A root of a word can be increased by prefix, suffix, proclitic, and enclitic letters. The letters of Arabic can be with or without diacritics and has various writing styles depending on the location of the letter in the word, i.e. in the beginning of the word, middle of the word, or in the end of the word [9]. The above features pose difficulties and challenges facing the applications based on Arabic language and particularly the Arabic information retrieval (AIR). For any information retrieval (IR) application, the language should be subject to preprocessing step. The significant processes in this step are the stemming, and stopwords and punctuation removal processes. There are two types of stemming used for AIR called heavy stemming and light stemming. Heavy stemming (also known as 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 AIR is believed to be better than the heavy stemming [10]. This is due to some words are generated from the same root but do not have similar meanings, therefore the root extraction process can unify the meanings of these words in an incorrect concept [1].

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 (Information Science Research Institute) stemmer is a newer and a root-extraction stemmer without a root dictionary. In other words, ISRI stemmer is similar to Khoja stemmer but does not validate roots against a dictionary. This feature makes ISRI stemmer more capable of stemming rare and new words [12]. Also, it differs from Khoja stemmer in that it returns a normalized form alif "ا" for different forms of Hamza "ء". [13]. Saad in [10] used a light stemmer for Arabic words derived from ISRI Arabic stemmer. This light stemmer consists of the following five major steps:

Step1: remove diacritics which representing Arabic short vowels.

Step2: remove length three and length two prefixes.
Step3: remove length three and length two suffixes.
Step4: remove connective waw "و" if it precedes a word beginning with waw "و"
Step5: normalize initial hamza "ء" to bare alif "ا".

Also, he used a reduction technique to increase of word matching chance called *morphAr*. The basic idea of *morphAr* technique is to merge both the light and heavy stemmers. If light stemming reduces the word form, then the light stem is returned, else, the root is returned.

In this paper, we proposed a combination of ISRI light and heavy stemmers depending on the presence of the word in the Arabic WordNet (AWN)[4]. Although the light stemmer gives more specific senses of a word than the heavy stemmer, but we don't always find the word that is formatted by the light stemmer in the AWN. This is the reason of why we need to combine the two stem methodologies. For example, the word 'الحاكمون' become 'حاكم' in light stemmer, which has 18 senses in the AWN while the original word become 'حكم' in heavy stemmer which has 62 senses in the AWN. However, we don't find the word 'رسول' in the AWN, that is extracted from the word 'الرسول' using the light stemmer while we find the word 'رسل' in the AWN, that is extracted from the word 'الرسول' using the heavy stemmer. Therefore, in the preprocessing process we first use the light stemmer for each word in the corpus and then check them in the AWN. If any word does not exist in the AWN, it should be altered by a rooted one using the heavy stemmer. Since this methodology depends on AWN, we called it as *AWNstem*.

2.2. Arabic WordNet Taxonomy

WordNet is a lexical database and is basically designed for the English language. It collects English words into sets of synonyms called synsets, gives short definitions and usage examples, and records a number of relations among these synonym sets. These relations include: hypernym, hyponym, coordinate terms, meronym, and holonym. After the great success of the English WordNet, many WordNets in several languages (such as Hebrew, Persian, African, Albanian, Indian, Arabic, etc.) are appeared to exist and they are all linked to the original English Wordnet [14]. So, when we talk about the Arabic WordNet (AWN), we actually mean Arabic-English thesaurus. This in turn can be mapped onto a number of other non-English thesauruses

[1] http://lucene.apache.org/core/features.html
[2] http:// lemurproject.org
[3] http://terrier.org

[4] Using the Arabic Wordnet (AWN v2) that is available in http://compling.hss.ntu.edu.sg/omw/

enabling translation on the lexical level to and from other languages [15]. Generally, AWN is a free lexical resource for modern standard Arabic constructed by the same methods developed for English WordNet [7]. It consists of 11,270 synsets and contains 23,496 Arabic expressions (words and multiwords) [14]. For example, the Arabic word "كلام" in the AWN has three synsets (senses) are labeled as language "لغة", manner_of_speaking "اسلوب التكلم", and speech "حديث". Each synset has a set of synonyms, the first synset, language, has synonyms which are {كلام، لُغة}, the second synset, manner_of_speaking, has synonyms which are {كلام أُسْلُوب، لهْجة}, and the third synset, speech, has synonyms which are {كلام،مُحادثة،مُخاطبة}. Sequentially, each synonym has also its own finite series of synsets and synonyms that form a taxonomy graph as shown in Figure (1):

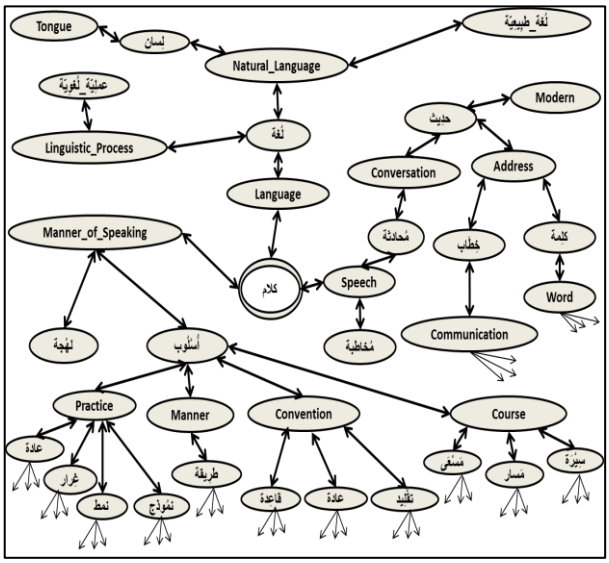

Figure (1): AWN taxonomy graph for Arabic word "كلام".

2.3 Semantic Similarity Measures

Semantic similarity measures have been widely used in natural language processing, word sense disambiguation, information retrieval, recommender system, question answering, information extraction, etc. Recently, the measures based on WordNet have attracted great concern. The semantic similarity measures based on WordNet have been grouped into four classes: path length-based measures, information content-based measures, feature-based measures, and hybrid measures [16]. Path length-based measures 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 [17]. Information content-based measures include *res*, *lin*, and *jcn* measures. In *res* measure, the similarity between two senses is related to their common information. The more two senses have in common, the more similar they are. It measures the common information as the information-content (IC) of the least common subsumer (LCS) [18]. LCS refers to the most specific sense that two senses share it as an ancestor in the WordNet hierarchy. The *lin* and *jcn* measures augment the IC of LCS of two senses with the sum of the information content of the individual senses. The *lin* measure scales the IC of LCS by this sum, while *jcn* subtracts the IC of LCS from this sum (and then takes the inverse to convert it from a distance to a similarity measure) [17].

In this paper we used the measure *path* as a path length-based measure to determine the similarity between any two Arabic senses in AWN. The shortest possible path occurs when the two senses are the same, in which case the length is 1. The *path*-based similarity formula is calculated as follows [18]:

$$Path_{sim} = 1/\text{shortest path length} \quad (2.1)$$

Where the path-length is measured in nodes (senses) with a note that the length of path between two members of the same sense (i.e., synonyms) is 1

In WordNet, two important relations links between the senses are the hypernym (superordinate) and hyponym (subordinate). For example, the English sense "speech", which is also a sense of the Arabic word "كلام", is a hyponym of the sense "auditory_communication". Conversely, the sense "auditory_communication" is hypernym of the sense "speech". The relations progression in ancestors or successors gets hypernym/ hyponym links as in Figure (2):

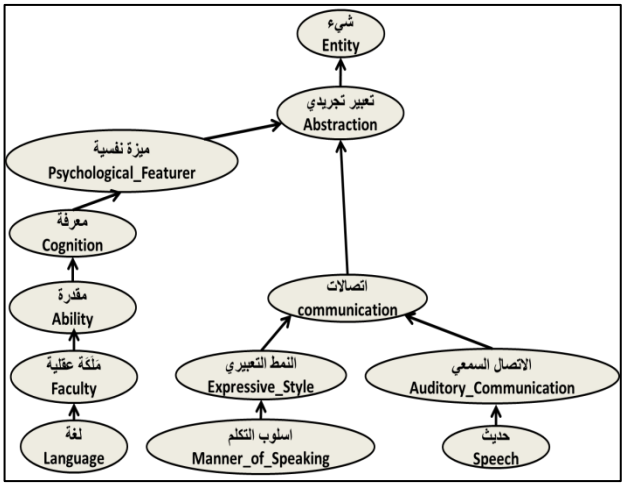

Figure (2): The hypernym/hyponym links in WordNet for senses of the Arabic word "كلام".

Now the path-length between “speech” and “manner_of_speaking”, for example, is 5 nodes according to Figure 2. The *path*-based similarity between “speech” and “manner_of_speaking” by Eq. (2.1) is calculated as follows:

$$path_{sim}(speech,manner\_of\_speaking) = 1/5 = 0.2$$

Another example, the *path*-based similarity between “speech” and "language" is calculated as 1/9=0.11.

The *path*-based similarity is just a measure for two senses. For measuring the semantic similarity for two words, i.e. by all their senses, we need to the so called "maximum path-based similarity". The maximum path- based similarity can be used to disambiguate the words which have multiple meaning after consider the neighboring words within the sentence (i.e., context). The maximum score resulted from the summation of the maximum similarities between the senses of an ambiguous word and the senses of all its neighbor words reveals the proper meaning of the ambiguous word within context.

Table 1: Disambiguate the word "حديث" in the first sentence

Senses of word "حديث"	Maximum similarity (with the word "تقنيات")	Maximum similarity (with the word "عصر")	Sum (total score)
recency	0.083	1.0	1.083
address	0.25	0.33	0.583
speak	0.143	0.167	0.310
talk	0.143	0.167	0.310
communicate	0.143	0.167	0.310
speech	0.125	0.143	0.268
conversation	0.111	0.125	0.236
modern	0	0	0
young	0	0	0
…			

For example, let we have the ambiguous word "حديث" which means a speech "حديث" in the AWN with synonyms are {"كلام","مُحادثة","مُخاطبة"} and means a recency "حداثة" with synonyms are {"عصْرِيّة","حداثة"} and means other senses in the AWN such as conversation, address, speak, etc. Let us also put this ambiguous word within two different sentences, for example:

1- The first sentence is "الكلام الطيب والحديث الحسن":
In the AWN, the word "كلام" has the following senses {speech, language, manner_of_speaking}, the word "طيب" has one sense {bouquet}, and the word "حسن" has senses include {beauty, better, approval, perfect,…etc.}.

2- The second sentence is "تقنيات العصر الحديث":
The word "تقنيات" has {technology, computer_technology, communications_technology, skill, craft,…etc.} senses and the word "عصر" has {era, day, tense, recency, afternoon, …etc.} senses.

Table 1 show that the best sense for the word "حديث" in the first sentence is "speech" while Table 2 show that the best sense in the second sentence is "recency

Table 2: Disambiguate the word "حديث" in the second sentence.

Senses of word "حديث"	Maximum similarity (with the word "كلام")	Maximum similarity (with the word "طيب")	Maximum similarity (with the word "حسن")	Sum (total score)
speech	1.0	0.125	0.2	1.325
conversation	0.5	0.111	0.167	0.778
address	0.143	0.125	0.333	0.601
speak	0.1	0.091	0.167	0.358
communicate	0.1	0.091	0.167	0.357
talk	0.091	0.083	0.143	0.317
recency	0.091	0.125	0.1	0.316
modern	0	0	0	0
young	0	0	0	0
…				

The sense "speech" of the word "حديث" in Table 1 is nearest to the sense "speech" of the word "كلام" with score 1.0 and nearest to the sense "bouquet" of the word "طيب" with score 0.125 and nearest to the sense "approval" of the word "حسن" with score 0.2. The total score of the sense "speech" is 1.325. It is the winner among other senses.

In Table 2, the sense "recency" of the word "حديث " is nearest to the sense "skill" of the word "تقنيات" with score 0.083 and nearest to the sense "recency" of the word "عصر" with score 1.0. The total score of the sense "recency" is 1.083. It is the winner among other senses.

3. Harmony Search

Harmony search (HS) is a stochastic evolutionary meta-heuristic algorithm developed by Geem et al. (2001), it is inspired from an artificial phenomenon, which is the musical harmony [19]. In HS algorithm, a musical instrument corresponds to a decision variable in optimization where its pitch range corresponds to a value range and a harmony corresponds to a solution vector [20].The aesthetic criteria in music corresponds the objective function that iteratively evaluates the solutions vectors [19]. Figure (3) show the HS algorithm [21],[22]

Harmony Search Algorithm

Initialize the HM with HMS randomly generated solutions.
Set generation count $g = 0$.
WHILE g < stopping criterion **DO**
*/*Generate a new solution*/*
FOR each decision variable **DO**
IF *rand(0,1)* < HMCR **THEN**
choosing any value from the HS memory
IF *rand(0,1)* < PAR **THEN**
choosing an adjacent value of the selected value.
END IF
ELSE
choosing a random value from the possible value range.
END IF
END FOR
IF new solution better than the worst solution in HM (in terms of fitness) **THEN**
Replace the worst solution in HM with new solution
END IF
Increment the generation count g = g + 1.
END WHILE
RETURN best solution

Figure (3): Harmony Search Algorithm

The HS algorithm has memory storage, called harmony memory (HM), consists of HMS harmony vectors. Harmony vectors and the corresponding values of objective function are both stored in the harmony memory matrix as follows [20]:

$$HM=\begin{bmatrix} x_1^1 & x_2^1 & \dots & x_N^1 & | & f(x^1) \\ x_1^2 & x_2^2 & \dots & x_N^2 & | & f(x^2) \\ \vdots & \vdots & \dots & \vdots & | & \vdots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_N^{HMS} & | & f(x^{HMS}) \end{bmatrix} \quad (3.1)$$

The first step of HS algorithm related to initialize the HM matrix by random variables and specifying the two basic parameters used for updating the HM. These parameters are the harmony memory considering rate (HMCR) in a process called HMC operation and the pitch adjusting rate (PAR) in a process called PA. HMCR indicates the rate of selecting value from the harmony. The value of this parameter usually varies from 0.7 to 0.99. If the value of HMCR is 0.95, for

example, then there is a more chance of constituting a new solution from the memory. PAR indicates the rate of modifying the value (that has been selected from the harmony memory) to one of the neighboring values. The value of this parameter usually varies from 0.1 to 0.5 [19]. The second step is to update the HM, a new harmony vector X^{new} is improvised by applying three rules: a memory consideration, a pitch adjustment, and a random selection. The decision variable x_k^{new} is generated by the memory consideration (i.e., is selected from any harmony vector *i* in 1, 2, ..., HMS) if a random number generated in the range (0, 1) is less than HMCR; otherwise, x_k^{new} is obtained by a random selection from between the search bounds. After finish the HMC operation, the PA operation can be selected with probability of PAR. In the PA operation, a selected solution value of decision variable from HMC operation is adjusted with upper or lower value. The PA rule has composite role in the HS algorithm. It is an exploration part for escaping from local optima, and it is also an exploitation part in the optimization process for finding exact optimal point by using fine tuning of decision variables [23]. If the new harmony vector X^{new} is feasible and better than the worst harmony vector X^w in HM, then X^{new} is included in HM and exclude the X^w from HM [20],[24].

4. Related Works

A few works have attempted to address the AIR in general and the Arabic query expansion in special. Abouenour et al. in [25] built an ontology using Arabic WordNet to expand the queries in order to improve the Query/Answer system. They indicated that the accuracy of answers was improved due to the utilization of the relations existing among the concepts of the ontology.

Mahgoub et al. in [5] introduced a system for addressing the semantic query expansion. The technique in their proposed system based on a domain independent semantic ontology constructed from Arabic Wikipedia. The system allows the user to either expands all terms in a single query or expand each term separately producing multiple queries. The system also can add terms from "Al-Raed" dictionary and from a constructed "Google_WordNet" dictionary. The system is tested on Zad-Al-Ma'ad corpus. They compared both the single expanded query and multiple expanded queries approaches against the traditional keyword based search. Both techniques were better than the baseline technique. While the multiple expanded queries approach performed better than the single expanded query in most levels.

Khafajeh et al. in [26] 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 documents from the National Computer Conference and 59 Arabic queries. Shaalan et al. in [27] suggest a method for query expansion on the AIR using Expectation Maximization (EM). EM is used to indicate similarity between two words based on their co-occurrence in a set of documents

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. Abbache et al. in [28] introduce three query expansion approaches named R2, RS, and RI derived from R1 approach that is the traditional baseline. R2 refers to automatic query expansion with all synonyms, RS refers to automatic query expansion with selected synonyms, and RI refers to interactive query expansion with selected synonyms. All the approaches depend on the Arabic WordNet (AWN). Their experiments show that any one of three approaches R2, RS, and RI results a significant increase in the recall measure, however, none of them increase the precision.

They stated that RS and RI have a little decrease in precision than R2 which a high decrease in precision due to the excessive number of synonyms for each word that are added from the AWN in one hand and the AWN's lack of universality of some of the words in the other hand. In this context, they report that with a good automatic selection of the right synonyms, the use of AWN for automatic query expansion improves the effectiveness of Arabic IR.

5. Proposed Sense-Based AIR System

In this work, we offer a new approach differs from the existed approaches in the other works and especially the mentioned in the section above and the most especially the most recent reference in [28]. It differs in that it automatically disambiguates each word (based on AWN) within all the documents in collection and the query before look at the query word synonyms and then automatically select (in a stochastic manner and based on candidate documents) the best synonyms of the query words. Therefore, the selection process is guaranteed to be within the correct sense in one hand and within the most candidate relevant documents in the other hand, and both these cases ensure the best quality. Also, the existing works don't focus on the delay that is taken online to test and find the best senses and best synonyms of the query words as well as the original search time of IR system. In fact, the growing in the search environment (i.e. increasing of test collection size) will inevitably leads to growing the overall delay.

The proposed system tries to improve Arabic IR (AIR) through query expansion in terms of efficiency and effectiveness. The proposed sense-based AIR system consists of three algorithms called "Maximum Path-Based Similarity Algorithm", "Proposed Harmony Search Algorithm", and "Expand Query & Matching Algorithm" respectively. Also, we called all these algorithms for the sake of brevity as "SPHS system", where the letter S in SPHS refers to use the sense in AIR while PHS refers to the proposed HS algorithm.

To illustrate the Algorithm1, Maximum Path-Based Similarity Algorithm, the following inputs should be present for the processing:

Let aw is an ambiguous word (associated with a *tf-idf* weight) that is required to be disambiguated though its context T.

$T = \{w_t | t = 1..n\}$: where w_t is a word in a text T associated with a *tf-idf* weight,

$aw^{Synsets} = \{aw^{Synset_i} | i = 1..s\}$: where aw^{Synset_i} is a set of synonyms within a sense i associated with ambiguous word aw,and the Synsets are extracted from the AWN.

$w_t^{Synsets} = \{w_t^{Synset_j} | j = 1..s\}$: where $w_t^{Synset_j}$ is a set of synonyms within a sense *j* associated with a word w_t, and the *Synsets* are extracted from the AWN. Algorithm1 is sown in Figure (4):

Algorithm1: Maximum Path-Based Similarity Algorithm

INPUT: $aw, T, aw^{Synsets}, w_t^{Synsets}$.
OUTPUT: $aw^{Synset_{best}}$ /*best sense of ambiguous word *aw* within a context T.*/
/*Best Sense Extraction*/
BEGIN
  FOR each aw^{Synset_i} in $aw^{Synsets}$ **DO**
    FOR $t = 1..n$ **DO**
      FOR each $w_t^{Synset_j}$ in $w_t^{Synsets}$ **DO**
        $sim_j \leftarrow path_{sim}(aw^{Synset_i}, w_t^{Synset_j})$
        $scores_j \leftarrow$ list(sim_j)
      END FOR
      $max_t \leftarrow$ max($scores_j$)
      $scores_t \leftarrow$ list(max_t)
    END FOR
    $sum_i \leftarrow$sum($scores_t$)
    $scores \leftarrow$ list(sum_i)
  END FOR
  best_sense← max($scores$)
END.

Figure (4): Maximum Path-Based Similarity Algorithm

To illustrate the Algorithm2, Proposed Harmony Search Algorithm, the following inputs should be present for the processing:

$G^{\varepsilon} = (V, E, S)$: refers to ε −neighborhood similarity graph.

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

D= $\{w_{d,j} | j = 1..m\}$: where $w_{d,j}$ is a word (associated with a *tf-idf* weight) in a document *D*

$S_q^{Synsets_{best}} = \{w_{q,i}^{Synset_{best}} | i = 1..n\}$:where $w_{q,i}^{Synset_{best}}$ is extracted using Algorithm1.

I_{max} = max iterations. IMP_{max} = maximum number of improvisations. VS= size of possible values. N= harmony size. HMS= harmony memory size. $HMCR$= harmony memory considering rate. PAR= pitch adjusting rate. BW= maximum pitch adjustment index (size of the movement).

The Algorithm2 is shown in Figure (5)

Algorithm2: Proposed Harmony Search Algorithm

INPUT: G^{ε}, Q, D, $S_q^{Synsets_{best}}$, I_{max}, IMP_{max} , VS , N, HMS, $HMCR$,PAR,and BW.
OUTPUT: doc^{best} /*best decision variable*/, $S_q^{synonym_{best}}=\{w_{q,i}^{synonym_{best}} | i = 1..n \}$ /*$w_{q,i}^{synonym_{best}}$ is the best synonym (associated with its *tf-idf* weight) of a word $w_{q,i}$*/
/*Search Mechanism*/
BEGIN
$HM[1][1..N]$ ← Set each decision variable with random value from 1 to collection size.
FOR j=1 to N **DO**
  Get $neighbors$ of $HM[1][j]$ by size of VS extracted from G^{ε}.
  FOR i=2 to HMS **DO**
    FOR k=1 to (HMS − 1) **DO**
      $HM[i][j] \leftarrow neighbors$[k]
    END FOR
  END FOR
END FOR
Calculate the fitness for HM $solutions$. Update scores of synonyms in $S_q^{Synsets_{best}}$.
FOR c=1 to I_{max} **DO**
  g = 0
  WHILE (g < IMP_{max}) **DO**
    FOR j=1 to N **DO**
      IF *rand(0,1)* < HMCR **THEN**
        choosing value from HM in a row i (i = 1,...,HMS).
        IF *rand(0,1)* < PAR **THEN**
          choosing an adjacent value of the selected value depend on BW.
        END IF
      ELSE
      choosing a random value from the possible value range from 1 to VS.
      END IF
    END FOR
    Calculate the fitness for the new solution. Update scores of synonyms in $S_q^{Synsets_{best}}$.
    IF new solution better than the worst solution in HM **THEN**
      Replace the worst solution in HM with the new solution.
    END IF
    g=g+ 1
  END WHILE
END FOR
RETURN decision variable has the maximum fitness and synonyms have the maximum scores.
END.

Figure (5): Proposed Harmony Search Algorithm

The neighbors of any decision variable in the harmony memory (HM) are extracted from the so called ε −neighborhood similarity graph, also called G^{ε} in short. In other words, the more similar documents (neighbors) of a searched document are constructed offline using G^{ε} after the vector space model (VSM) of the document collection has been completed. The ε is a threshold parameter controls the number of links in the graph. Let $G = (V, E, S)$ is an undirected weighted graph for a document collection in which V is a set of documents in VSM, 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} equals to the cosine similarity. $G^{\varepsilon} = (V, E, S)$ is the ε −neighborhood similarity graph in which each edge e_{ij} in the graph can connect two documents d_i and $d_j$with weight s_{ij} represents the cosine similarity if and only if the cosine similarity result ≥ ε.

Initially the decision variables (documents) in the first solution (harmony) are randomly chosen in an integer interval from 1 to the collection size. The other solutions are extracted, using G^{ε} graph, from the neighbors of documents in the first solution. For example, let a document d_i has 1500 neighbor documents, these documents are ordered according to the similarity of d_i. If VS =1000, this means that the possible value range for the search process is closed in 1 to 1000 strong neighbor documents while the other 500 weak neighbors are ignored. Also, if HMS=100, as an example, then the range from 1 to 100 of the harmony memory is a subrange taken from the strong documents ranged from 1 to 1000. The search space structure using G^{ε} improves the convergence speed of harmony search algorithm. The fitness function in the algorithm is the cosine similarity between a query and a document (refer to a decision variable). After obtained the fitness for each document in the solution, the solution fitness is just the maximum fitness obtained by a document within the solution. 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 (definition) of that synonym, and the query is the context where we need to find 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. An update of the synonyms scores 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, a number of new solutions are generated. A decision variable is updated after check HMCR. If a random value in range (0,1) is lesser than HMCR, then the location of candidate document is chosen from the current harmony memory HM within the range from 1 to HMS, otherwise it chosen from the original neighbor documents within the possible value range from 1 to VS. In the first case, if a random value in range (0,1) is lesser larger than PAR, then the location of the new document is the same as the candidate document, otherwise the location of the new document is chosen depending on probability of flipping a coin that just determine the movement direction, down/up, away from the candidate

document. For example, if BW is the size of the movement, the $candidate_{index}$ is the location of the candidate document within the range from 1 to VS, then the location of the new document in the down movement will be indexed as $\{candidate_{index} - randint\ (0, min(BW, candidate_{index}))\}$ and indexed in the up movement as $\{candidate_{index} + randint\ (0, min(BW, VS - candidate_{index} - 1))\}$. After a predefined number of iterations, the best solution is memorized and the final output of the algorithm is the document that has maximum fitness (i.e. the best document), and a set of synonyms have maximum scores (i.e. the best synonyms). The best document is again subject to another process while the best synonyms are weighted
by tf-idf weights and inserted into the original query to obtain expanded query as in Algorithm3 in Figure (6):

Algorithm 3: Expand Query & Matching Algorithm

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
Add the synonyms in $S_q^{Synonym_{best}}$ to query, name the result as $Q^{expanded}$.
Add doc^{best} to its $neighbors$ list, name the result as $docs^{candidate}$.
Calculate the cosine similarity between $Q^{expanded}$ and $docs^{candidate}$ with consider the matching of words and senses.
Sort the results in descending order.
END.

Figure (6): Expand Query & Matching Algorithm

6. Experimental Results and Discussion

The proposed system is experimented on ZAD corpus. For the IR systems that deal with large-scale databases, the retrieval efficiency become1975s a critical task. In this paper the efficiency was in the position of interest. we tend to draw a simple simulation to a client-server database structure. The so called Redis[5] is a database server that provides a very simple client protocol similar to Telnet [29]. The databases used in the proposed and traditional systems are stored in separate places within the Redis server include the Inverted-Index, VSM, and G^{ε}. 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 compared to the traditional baseline algorithm. The traditional baseline algorithm used with no WSD mechanism and depends on *morphAr* stemmer technique. Table 3 shows the average response time, the average of precision and recall @ 10, and the MAP measure for a sample of ten queries in ZAD corpus.

Table 3: Average performance evaluation with a ZAD collection of a sample of ten queries.

ZAD Average Performance	Traditional	SPHS
Avg. Response Time (sec./query)	0.971	0.894
Avg. precision	0.390	0.510
Avg. recall	0.235	0.322
MAP measure	0.5815	0.7762

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 sees the first relevant document is a more important than the second in the ranked list and so on. Depending on MAP measure entered in Table 3, the SPHS system outperforms the traditional baseline system which has no WSD technique in a rate of 19.5% and reducing of latency in an approximate rate of 0.077 second for each query. Figure (7) shows the 11-point interpolated recall-precision curves for 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.

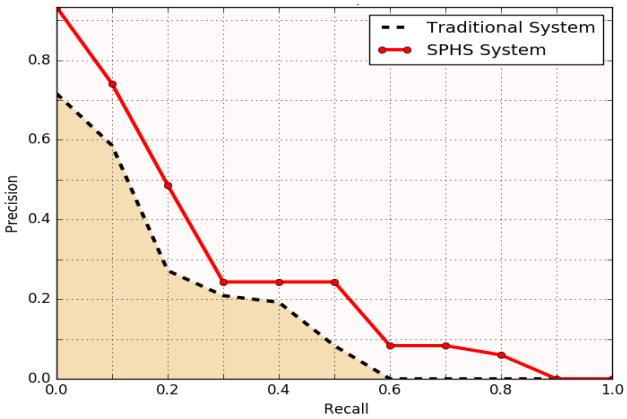

Figure (7): Average recall-precision curves for the sample of ten queries

The proposed HS algorithm in SPHS is augmented by an efficient initial search space structure using the similarity graph G^{ε} that improves the convergence speed of harmony search algorithm. Therefore, for a few size parameters, the algorithm converges in a reasonable time. The HS parameters are set as follows: I_{max}=5, IMP_{max} =10, VS = 1000, N=10, HMS =10, $HMCR$=0.8, PAR=0.25, and BW=5.

It is clear that the tested response time in Table 3 shows the superiority in the efficiency of the proposed system over the traditional system. Superiority in the proposed system efficiency is directly proportional to

[5] http://redis.io

the size of inverted index and sizes of inverted lists (posting lists) within inverted index in traditional systems. In other word, 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 hashing access. Also in large-scale environment, it is expected that the majority of the inverted lists will have larger sizes. Therefore, our system is more efficient for dealing with an environment has a huge number of documents than an environment has a few documents. We also noticed, through our experiments on ZAD corpus, that increasing of the number of visited documents is not a prerequisite to know the amount of delay. Real delay is also caused by the size of the document (i.e., number of words in the document) that travels from the server to the client. For example, in ZAD corpus, the document numbered 480 have only one word that delays 0.000185 second while the document numbered 2256 have 567 words that delays 0.006094 second. Accurately, for the tests conducted in Table 3, we find that the number of visited documents in SPHS system is 984 out of (2730) documents, while in the traditional algorithm is only 563. This explains that SPHS system has more chance to search different sizes of documents because it can reach documents are prohibited due to of the influence of words' weights. It should also be noted that the effectiveness may be 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.

7. Conclusion

In this paper, we developed an intelligent technique uses Harmony Search (HS) algorithm to improve the sense-based Arabic information retrieval. HS is used to search the best synonyms for the query expansion and the most relevant documents simultaneously. The system is compared with traditional baseline system which has non-expanded query. ZAD Arabic corpus is used to test the system performance. The query expansion increases the effectiveness while the stochastic optimization search of HS algorithm increases the efficiency. The experimental results exhibit the performance superiority of the proposed system over the traditional system in terms of the precision, recall and latency

References:

[1] M. K. Saad and W. Ashour," Arabic Morphological Tools for Text Mining". Published at the 6th International Conference on Electrical and Computer Systems (EECS'10), Lefke, Cyprus, 112-117, 2010.

[2] M. Shaheen, A.M. Ezzeldin, "Arabic Question Answering: Systems, Resources, Tools, and Future Trends ", King Fahd University of Petroleum and Minerals, Arab J Sci Eng 39, 4541–4564, 2014.

[3] 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, 4(2), 722-726, 2014.

[4] T. Rachidi, M. Bouzoubaa, L. Elmortaji, B. Boussouab, and A. Bensaid, "Arabic user search query correction and expansion," in Proc. 1st Plenary Inf. Technol. Pole of Competences Conf., 2003.

[5] A. Y. Mahgoub, M.A. Rashwan, H. Raafat, M.A. Zahran, and M.B. Fayek, "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,87-92, 2014.

[6] Alia Karim A. Hassan and Mustafa J. Hadi, "Proposed MABC-SDAIR Algorithm for Sense-Based Distributed Arabic Information Retrieval", Journal of Theoretical and Applied Information Technology, 95(3), 543-553, 2017.

[7] S. Elkateb, W. Black, P. Vossen, D. Farwell, H. Rodríguez, A. Pease, M. Alkhalifa, "Arabic WordNet and the Challenges of Arabic", The Challenge of Arabic for NLP/MT. International conference at the British Computer Society, London,15-24, 2006

[8] M.N. El-Gedawy."Using Fuzzifiers to Solve Word Sense Ambiguation in Arabic Language". international Journal of Computer Applications, 79 (2),1-8, 2013.

[9] A. Abu-Errub, A. Odeh, Q. Shambour, O. Al-Haj Hassan ," Arabic Roots Extraction Using Morphological Analysis". IJCSI International Journal of Computer Science Issues, 11(2), 128-134, 2014.

[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, A. Al-Ibrahim, and R. Al-Mutawa, "A Comparative Study on Arabic Stemmers". International Journal of Computer Applications, 125(8), 2015.

[12] A.M. Ezzeldin, M. Shaheen, "A survey of Arabic question answering: challenges, tasks, approaches, tools, and future trends" In: The 13th International Arab Conference on Information Technology, 280-287, 2012.

[13] M. El-Defrawy, Y. El-Sonbaty, N.A. Belal, "Enhancing Root Extractors Using Light Stemmers", 29th Pacific Asia Conference on Language, Information and Computation: Posters, Shanghai, China, 157-166, 2015.

[14] H. Ishkewy, H. Harb, and H. Farahat, "Azhary: An Arabic Lexical Ontology", International Journal of Web & Semantic Technology (IJWesT), 5(4), 71–82, 2014.

[15] L. Abouenour, K. Bouzoubaa, P. Rosso, "Improving Q/A using Arabic Wordnet", In: Proceedings of the Arab Conference on Information Technology(CIT'08), IBTIKARAT Research Group, Tunisia, 2008.

[16] L. Meng, R. Huang, and J. Gu,"A Review of Semantic Similarity Measures in WordNet", International Journal of Hybrid Information Technology 6(1), 1-12, 2013.

[17] T. Pedersen, S. Patwardhan, J. Michelizzi , "WordNet: Similarity - Measuring the Relatedness of Concepts", In: AAAI 2004, San Jose, CA, 1024–1025, 2004

[18] H. Chen, "String Metrics and Word Similarity applied to Information Retrieval", M.Sc. Thesis, University of Eastern Finland School of Computing, 2012.

[19] S. A. Abed, S. Tiun, N. Omar, "Harmony Search Algorithm for Word Sense Disambiguation", PLoS One, 10(9),1-17, 2015.

[20] B. Wu, C. Qian, W. Ni, and S. Fan, "Hybrid harmony search and artificial bee colony algorithm for global optimization problems", Computers & Mathematics with Applications, 64, 2621-2634, 2012.

[21] K. Mohanarangam, R. Mallipeddi, "Harmony Search Algorithm with Ensemble of Surrogate Models," in 2nd International Conference on Harmony Search Algorithm, 19-28, 2015.

[22] X. Wang, X. Gao, and K. Zenger, "An Introduction to Harmony Search Optimization Method", SpringerBriefs in Applied Sciences and Technology, Chapter 2, 2015.

[23] J. H. Kim, H. M. Lee, and D. G. Yoo," Investigating the Convergence Characteristics of Harmony Search", Advances in Intelligent Systems and Computing, Springer Verlag, 382 , 3-10, 2016.

[24] N. Nahas and D. Thien-My, "Harmony search algorithm: application to the redundancy optimization problem", Engineering Optimization, 42(9), 845–861, 2010.

[25] L. Abouenour, K. Bouzoubaa, P. Rosso. Improving Q/A using Arabic wordnet. InProc. The 2008 International Arab Conference on Information Technology (ACIT'2008), Tunisia, 2008.

[26] H. Khafajeh, N. Yousef, and G. Kanaan, "Automatic Query Expansion for Arabic Text Retrieval Based on Association and Similarity Thesaurus", Proceedings of EMCIS, 1-17,2010.

[27] K. Shaalan, S. Al-Sheikh , and F. Oroumchian, "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, 167-176, 2012.

[28] A. Abbache, F. Meziane, G. Belalem, F. Z. Belkredim," Arabic Query Expansion Using WordNet and Association Rules" International Journal of Intelligent Information Technologies, 12(3), 51-64, 2016.

[29] A. Chinnachamy, "Instant Redis Optimization How-to", Packt Publishing, Paperback: 1-56, 2013.