A New Semantic Distance Measure for the VSM-Based Information Retrieval Systems

Aya M. Al-Zoghby

Abstract One of the main reasons for adopting the Semantic Web technology in search systems is to enhance the performance of the retrieval process. A semantic-based search is characterized by finding the contents that are semantically associated with the concepts of the query rather than those which are exactly matching the query's keywords. There is a growing interest in searching the Arabic content worldwide due to its importance for culture, religion, and economics. However, the Arabic language; across all of its linguistics levels; is morphologically and syntactically rich. This linguistic nature of Arabic makes the effective search of its content be a challenge. In this study, we propose an Arabic semantic-based search approach that is based on the Vector Space Model (VSM). VSM has proved its success, and many studies have been focused on refining its old-style version. Our proposed approach uses the Universal WordNet (UWN) ontology to build a rich index of concepts, **Concept-Space** (CS), which replaces the traditional index of terms, **Term-Space** (TS) and enhances the Semantic VSM capability. As a consequence, we proposed a new incidence indicator to calculate the **Significance Level of a Concept** (SLC) in the document. The new indicator is used to evaluate the performance of the retrieval process semantically instead of the traditional syntactic retrieval that is based on the traditional incidence indicator; **Term Frequency** (TF). This new indicator has motivated us to develop a new formula to calculate the **Semantic Weight of the Concept** (SWC). The SWC is necessary for determining the **Semantic Distance** (SD) of two vectors. As a proof of concept, a prototype is applied on a full dump of the Arabic Wikipedia. Since documents are indexed by their concepts and, hence, classified semantically, we were able to search Arabic documents efficiently. The experimental results regarding the Precision, Recall, and F-measure presented a noticeable improvement in performance.

A.M. Al-Zoghby (✉)
Faculty of Computers and Information Systems, Mansoura University, Mansoura, Egypt
e-mail: aya_el_zoghby@mans.edu.eg

K. Shaalan et al. (eds.), *Intelligent Natural Language Processing: Trends and Applications*, Studies in Computational Intelligence 740,
https://doi.org/10.1007/978-3-319-67056-0_12

Keywords Semantic search systems · Vector space model (VSM)
Universal wordnet (UWN) · Concept-space (CS) · Significance level of concept (SLC) · Semantic weight of concept (SWC) · Semantic distance (SD)
Arabic language

1 Introduction

The ambiguity of the search query's keywords is one of the main problems that may frustrate the search process efficiency. The use of the terminological variations for the same concept, Synonyms, creates a many-to-one ambiguity. Whereas, the use of the same terminology for different concepts, polysemous, creates a one-to-many ambiguity [1, 2]. The problem becomes more sophisticated with a highly ambiguous language such as Arabic [3, 4]. For example, the optional vowelization in modern Arabic text increases the polysemy of its written words [5, 6].

Traditionally, the search engines are characterized by trading off a high-recall for low-precision. This is caused mainly due to their sensitivity to the query keywords, and the misinterpretation of the synonymous and polysemous terminologies [7]. In other words, not only all relevant pages are retrieved, but also some other irrelevant, which directly affects the Precision. Moreover, the absence of some relevant pages is leading to low Recall. A recommended solution is to use the *semantic search*, which relies on ontological resources for semantic indexing instead of the lexical indexing that are commonly used by traditional search systems. Thus, the *Semantic search* aims to resolve the semantic ambiguity by retrieving the pages referring to the *semantic interpretation*, hence a particular *concept*, of the search query instead of the pages that are just mentioning its keywords [8, 9].

This research proposes an enhanced semantic VSM-based search approach for Arabic information retrieval applications and the like. In the proposed search approach, we built a concept-space which is used to construct the VSM index. This model enabled us to represent the Arabic documents as semantic vectors, in which the most representative concepts are got the highest weights. This representation allows a semantic classification for the search space. Thus, the semantic retrieval abilities, reflected in its Precision and Recall values, can be obtained. The evaluation of the retrieval effectiveness using the concept-space index resulted in a noticeable improvement in terms of the Precision and the Recall as compared to the traditional syntactic term-space baseline.

The rest of the paper is organized as follows: Sect. 2 describes the main aspects of the proposed model. Section 3 represents the architecture of the proposed model and the implementation aspects. A system is implemented, and the experimental results are discussed in details in Sect. 4. Finally, the paper is concluded at the last section. The list of the algorithms developed to implement the proposed system are listed in the article's appendix.

2 The Proposed Approach

This research proposed an enhanced semantic VSM-based approach for Arabic information retrieval applications and the like. The VSM is a conventional information retrieval model that has demonstrated its ability to represent documents in a computer interpretable form [10].

In the proposed model, we built a rich VSM index of concepts, concept-space (CS) that is enabling the representation of the documents as semantic vectors, in which the most relevant concept are given the highest weights. This semantic representation allows a semantic classification of the documents. Thus the semantic search facilities can be obtained. The construction of CS is derived from the semantic relationships obtainable form the UWN. UWN provides a corresponding list of meanings and shows how they are semantically associated [11]. Fortunately, the UWN supports the Arabic language and its dialects as well. As a proof of concept, a system is implemented on the Arabic Wikipedia. The evaluation of the system's semantic retrieval effectiveness is tested in terms of the Precision and Recall. It resulted in noticeable improvements in its performance as compared to the syntactic term-space based systems.

The key contributions of the study, and how it is distinguished from the traditional VSM are highlighted at the next sections.

2.1 A Novel Indexing Approach

In Semantic Web, terms are used to explain concepts[1] and their relations, [8, 9]. Consequently, the concepts sharing some terms in their definition will share many perspectives of their meanings. This can be realized when concepts can be identified and used as a semantic index of the VSM. For performance evaluation purposes, we produced three indices: Term-Space (TS), Semantic Term-Space (STS), and Concept-Space (CS). As specified by Definition 1, each entry of TS considers all inflected forms of the term. For more clarification, see the TS block of Fig. 1.

Each entry of the STS dictionary, on the other hand, is the UWN semantic expansion of the corresponding TS entry. In other words, each term in the TS is represented by its semantic expansions[2]; as specified by Definition 2[3] and the STS block of Fig. 1.

However, the generation of STS index has revealed some drawbacks that need to be addressed. It might produce duplicated entries that are *directly* or *indirectly*

[1]In this paper, whenever the word *term* is used; it refers to a single word. In the VSM, this term is an entry of the TS. Likewise, whenever the word *concept* is used; it refers to a single concept that is defined in terms of the set of related *terms*, and represented by a single entry in the CS.

[2]More precisely, the set of related inflected forms of its semantic expansions.

[3]More details about producing STS index are can be found in [12], and [13].

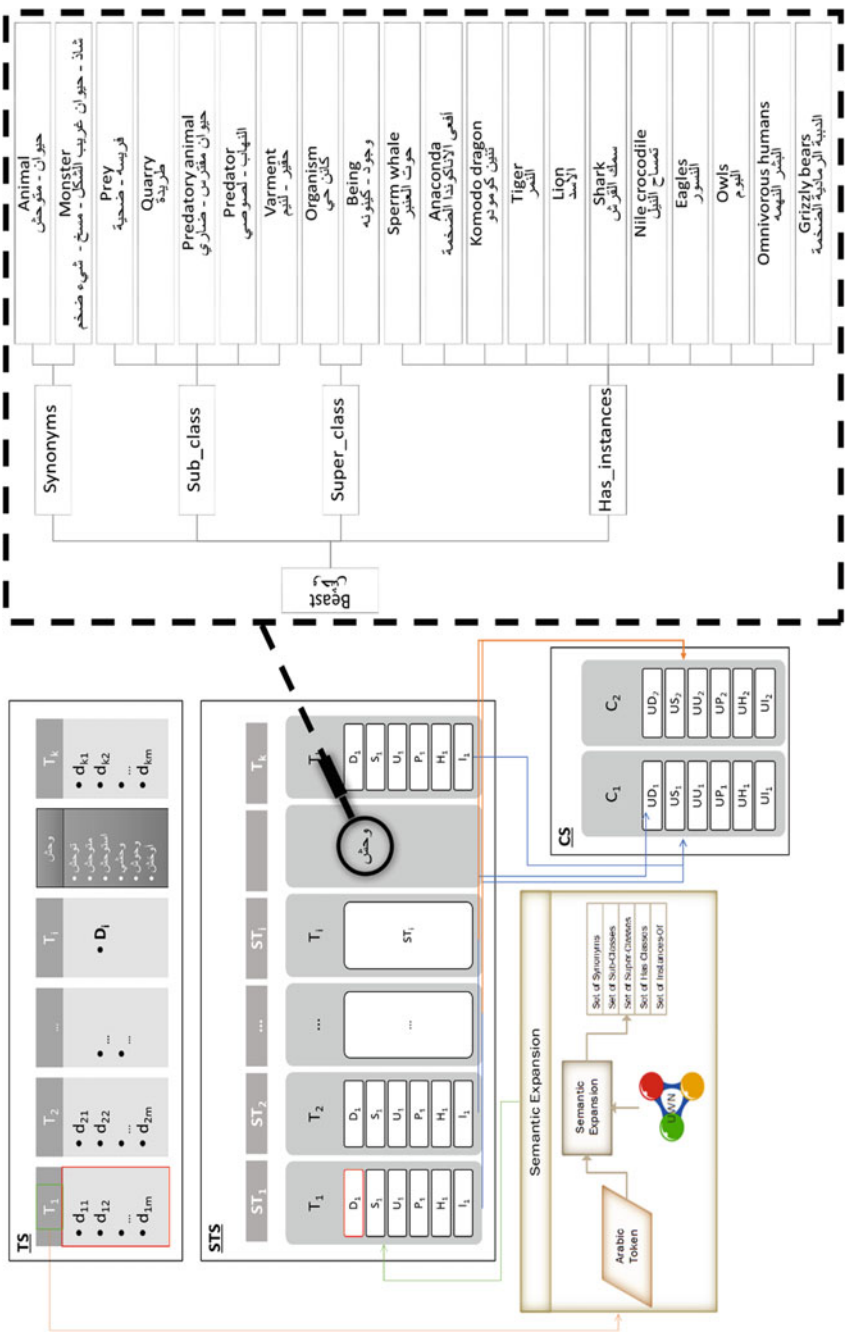

Fig. 1 TS, ST, and CS entries example

linked, see Fig. 2. This duplication causes a kind of semantic redundancy that should be resolved to improve the retrieval performance.

For example, as presented in Fig. 2, the terms:

'Beast[4]/وحش', 'Savage[5]/متوحش - بدائي - همجي', 'Brute[6]/ بهيمةشخص وحشي - بهيمي -','Behemoth[7]/ شخص ضخم جدا', and 'Demon[8]/ شخص ذو قوة - عفريتالروح الحارسة - شيطان'

are all semantically expanded entries at the STS index. However, there are some direct and indirect redundancies at these expansions. This redundancy is presented at the existence of terms that share their synonyms with other terms or even with the synonyms of other terms. For example, the synonyms '*Beast'* of the terms '*Savage'* and '*Brute'* with the term '*Beast'* itself. Moreover, the shared synonym '*Wildcat'* of both terms '*Savage'* and '*Brute'*. Also, the synonym *'Monster'* of '*Beast'* that is shared by both *'Behemoth'* and *'Demon'*.

To overcome these deficiencies of the STS, we introduced a novel indexing approach to build a CS index that is capable of improving the search performance. As stated at Definition 3, an entry of the CS dictionary is a concept. In our study, the concept is defined by: a main keyword identifying the concept, all of the terms enclosed by the concept's meaning, all of these terms morphological derivations, and all of their UWN semantic expansions. See the CS block of Fig. 1.

The generation and the movement from one indexing type to the next advanced type are depicted at the '*Morphological Indexing'* and '*Semantic and Conceptual Indexing'* phases at the system architecture presented in Fig. 7.

As VSM has proved its capability to represent documents in a computer interpretable form, we tried to improve its performance by replacing its traditional index TS with the new index CS, see Fig. 3. The CS index enabled us to represent documents as semantic vectors, in which the highest weights are assigned to the most representative concept. Therefore, the vector is accurately directed if its highest weight is assigned to the document's fundamental concept, Fig. 3. Thus the semantic search facilities, reflected in its Precision and Recall values, can be gained [4].

2.2 The Significance Level of a Concept (SLC)

In the literature, the performance evaluation of the retrieval capability of indices is usually measured using the following measures: *Document Frequency (df)*, *Term*

[4]http://www.lexvo.org/uwn/entity/eng/beast.

[5]http://www.lexvo.org/uwn/entity/eng/savage.

[6]http://www.lexvo.org/uwn/entity/s/n9845589.

[7]http://www.Lexvo.Org/uwn/entity/s/n10128909.

[8]http://www.lexvo.org/uwn/entity/eng/demon.

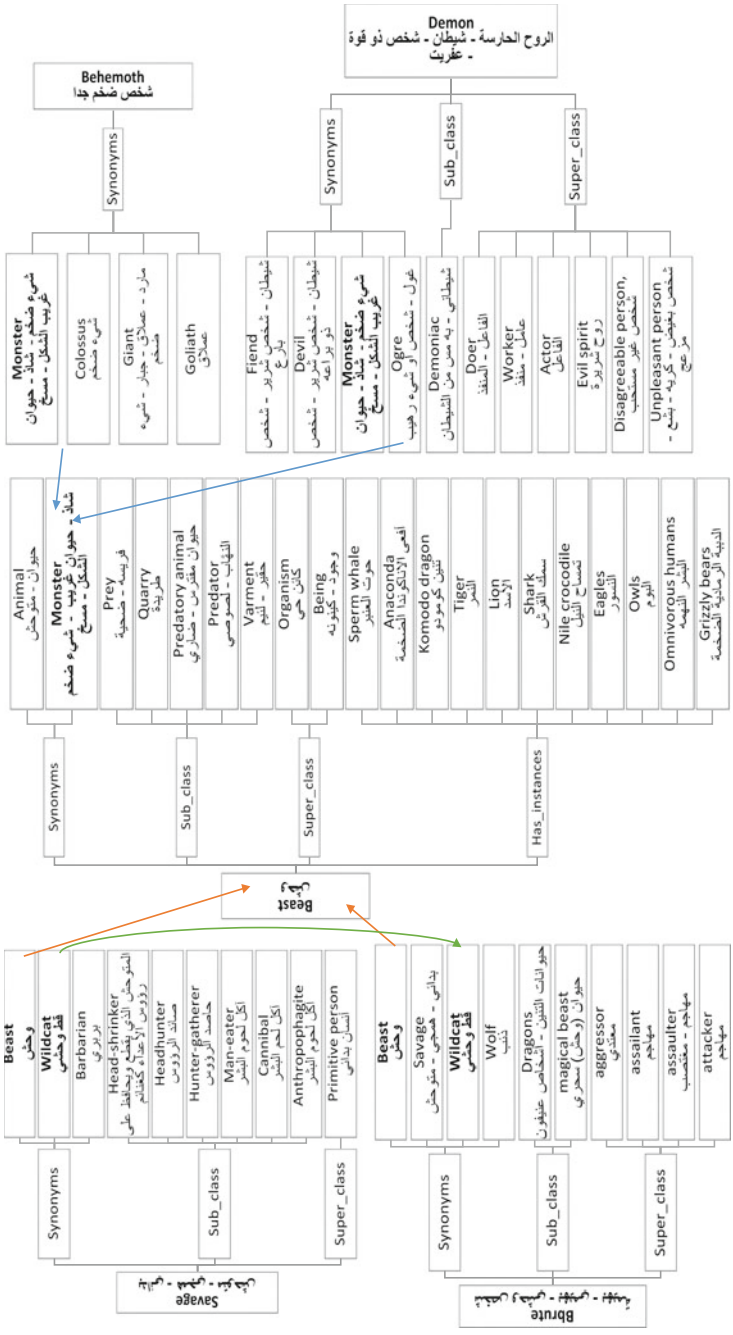

Fig. 2 The semantic expansions redundancy at STS. The semantic expansions presented at the figure are captured from UWN

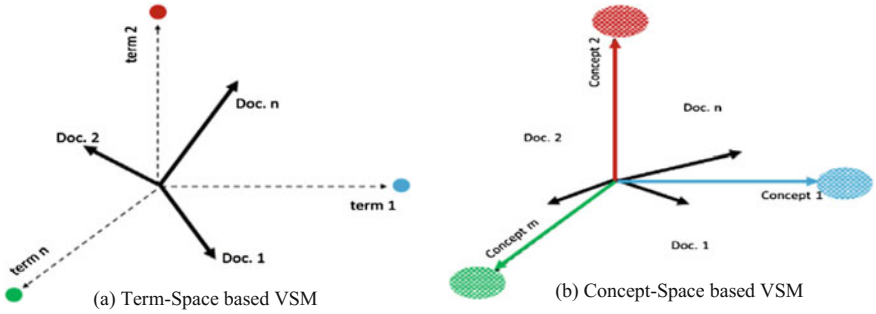

Fig. 3 The enhancement of the traditional VSM to be conceptual VSM

Frequency (tf), and *Weight* (*w*). The bigger the value of the *df, tf, or w* for an index entry, the more relevant documents are found. It is obvious that following the traditional method of calculating the occurrences of the term or its semantic expansions across the document is neither fair nor efficient. It might cause deceptions since documents are considered as relevant basing on the absolute frequency of terms. Therefore, the calculation of the frequency must be controlled by other factors that consider the *relevance degree* instead of the *absolute frequency*. As a matter of fact, the direct increment of *df*, *tf*, and *w* for each occurrence of the term itself, its semantic expansions, or its conceptualization terms, respectively, may suffer from inaccurate results since there are variations in the relevance levels the expansions. This has motived us to introduce a stage of processing that calculates the *significance level* of the term/concept as a more accurate alternative of the traditional *term frequency* which positively impact the recognition of relevant documents.

In VSM, the weight of the term *t* in document *d* refers to the term's capability of distinguishing the document. Traditionally, the weight *w* of *t* in *d* is defined in terms of its frequency *tf*, which is the number of times that *t* occurs in *d*. However, when the semantic conceptual model is adopted, the equation that calculates the weight will no longer be accurate, and three factors, which are affecting the calculation of the decisive weight, should be taken into consideration.

Definition 1 Term-Space (TS) The TS is defined as the set of all distinct terms belonging to the knowledge source[9] as follows:

$TS = \{T_1, T_2, \ldots, T_i, \ldots, T_k\}$, where:

T_i is a set of inflected forms of Term #i at the TS, i.e. $T_i = \{t_{i1}, t_{i2}, \ldots, t_{ij}, \ldots, t_{im}\}$

k = # terms in TS

m = # inflection forms of Term i

[9]The used knowledge source is the AWDS, which stands for Arabic Wikipedia Documents Space.

Definition 2 Semantic Term-Space (STS) Given TS; the STS is defined as follows:

$STS = \{ ST_1, ST_2, \ldots, ST_i, \ldots, ST_k \}$, where:

ST_i = Semantic_Expansion (T_i) $\cup$ T_i[10]

Semantic_Expansion (T_i) = $\{S_i, U_i, P_i, H_i, I_i\}$

Thus,

$ST_i = \{e_{i1},\ldots, e_{ix,}, \ldots, e_{in}\}$[11]

e_{ix} = the inflectional/semantic expansion #x of the term T_i

n = the total count of T_i sematic expansions

Definition 3 Concept-Space (CS) Given STS, the CS is generated from groups of concepts, each of which is represented by interrelated semantic terms of the STS as follows:

$CS = \{C_1, C_2,\ldots, C_q\}$, where:

$C_i = \{ST_{i1}, ST_{i2}, \ldots, ST_{ij}, \ldots, ST_{ic}\}$[12]

Thus,

$C_i = \{e_{11}, \ldots, e_{1x,}, \ldots, e_{1n}, \ldots, e_{c1},\ldots, e_{cx,}, \ldots, e_{cn}\}$

q = # concepts extracted from the STS

C_i = the concept #i, which is defined by a set of semantic terms from STS

c = # semantic terms used to define the concept #i

First, the term needs to be matched against all of its **inflected forms** that occur in *d* rather than the input or the normalized form. Therefore, not all matches would have the same weight.

Second, the semantic expansions of terms can be classified into five different types or relations: *Synonyms, SubClasses* and *HasInstances (Specialization expansions)*, and *SuperClasses* and *InstancesOf (Generalization expansions)*. It is evident that the Synonyms expansions are **semantically closer** to the original term than either its generalized or specialized expansions. Moreover, the SubClasses and SuperClasses expansions are worthier than those of HasInstances or InstancesOf, since the former types represent classes that encompass a set of related concepts while the latter types represent instances referring to specifically related elements.

---

[10]See Ti at Definition 1.

[11]Extracted from the UWN.

Si = {s1,…,sa}, //set of Synonyms

Ui = {u1,…,ub}, //set of Sub-Classes

Pi = {p1,…,pc}, //set of Super-Classes

Hi = {h1,…,hd}, //set of Has-Instances

Ii = {i1,…,ie}, //set of Instances-Of

All of these expansions can be accumulated in the set of all expansions on the term Ti: {ei1, …..ein}.

Each expansion of s, u, p, h, & i, is represented as a pair of the expansion-word and the expansion-confidence as (word, conf.).

[12]Therefore, each concept C is defined in terms of all expansion sets of each ingredient STs. I.e., the accumulation of the subsets {ei1, …, eix,, …, ein}.

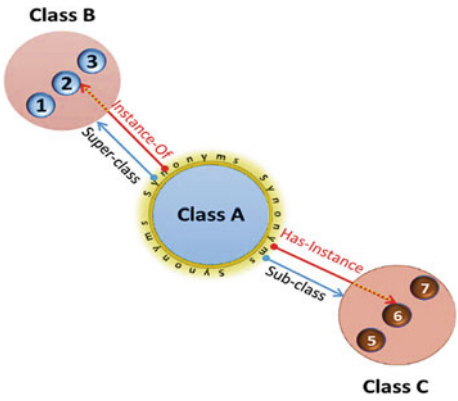

Fig. 4 Synonyms, sub-classes, super-classes, has-instance, and instance-of relationships

This also means that matches with different **semantic relations** also have different **distances** that directly affect the weights.

Third, the available semantic expansions of each semantic relation type have different **confidences** because not all of expansions are relevant to the original term with the same degree or weight. So, an additional confidence factor should be used when calculating the weight.

The factors above impede the term frequency to be used as an accurate *incidence indicator* for the semantic expansions in the semantic search process. Hence, we introduced a new *incidence indicator* of the term t, and consequently the concept c that is defined in terms of t, in document d. This indicator is based on the *significance level* of the term in the document instead of its frequency count. The new measurement, *significance level*, is computed in terms of the *association strength* of each expansion of the term. The *association strength* depends on the *distance* of the semantic relation and the *confidence* of the semantic expansion. The distance of the semantic relation is a constant[13] that is heuristically determined by the semantic closeness of the expansion as declared before. The value of the confidence, on the other hand, is directly determined by the UWN specifications.

Figure 4 shows the semantic closeness of the expansion. As depicted in this figure, the set of ***Synonyms*** has an exact matching to the original term. Therefore, we decided to set the value of the multiplying coefficient of synonyms to be 1. On the other hand, the ***Sub/Super classes*** represent some degree of generalizations such that they are not found by the exact match with the original term. However, they are more closely related than ***instances*** since they indicate a general perspective of the meaning. Whereas the instances represent certain items that may share one or more characteristics of the original term's meaning. Therefore, we decided to set the value of the multiplying coefficient of the sub/super classes to be

[13]Multiplying coefficient along the distance scale.

0.75 and that of the has-instances/instances-of classes to be 0.5. More illustrative examples can be found in [14]. Formally, let *AS(e)* denotes the *Association Strength* of the expansion *e*. As presented at Eq. (1), it is defined in terms of the *confidence* of the expansion *e*, and the *distance* of its semantic relation type. The new incidence indicator, which denotes the *Significance Level of a Term ST* at document *d*, is defined by *SLT(ST,d)* as presented at Eq. 2. Where *n* is the number of all expansions *e* of term *ST*.[14] The $\mathrm{tf}(e_x)$ factor represents the frequency of expansion e_x instances occurred in document *d*.

$$\mathrm{AS}(e) = \mathrm{confidence}(e) * \mathrm{distance}(e) \tag{1}$$

$$\mathrm{SLT}(\mathrm{ST}, d) = \sum_{x=1}^{n} \mathrm{AS}(e_x) * \mathrm{tf}(e_x) \tag{2}$$

Let SIDF(ST) denotes the Semantic Inverse Document Frequency of the term ST. It is defined using Eq. 3.[15,16] Thus, the Semantic Weight of term ST in document d is defined by SW(ST,d) presented at Eq. 4. In terms of concepts, the Significance Level of a Concept (SLC) is defined by Eq. 5. Where c is the count of the semantic terms used to define the concept C.

$$\mathrm{SIDF}(\mathrm{ST}) = \log \frac{|D|}{|\{e_x \in d \mid e_x \in \mathrm{ST}\}|} \tag{3}$$

$$\mathrm{SW}(\mathrm{ST}, d) = \mathrm{SLT}(\mathrm{ST}, d) * \mathrm{SIDF}(\mathrm{ST}) \tag{4}$$

$$\mathrm{SLC}(C, d) = \sum_{i=1}^{c} \mathrm{SLT}(\mathrm{STi}, d) \tag{5}$$

Finally, let SIDF(C) denotes the Semantic Inverse Document Frequency of the concept C as defined by Eq. 6.[17] The Semantic Weight of each concept in the new Concept Space, SW(C,d), is defined by Eq. 7.

$$\mathrm{SIDF}(C) = \log \frac{|D|}{|\{e_x \in d \mid e_x \in C\}|} \tag{6}$$

$$\mathrm{SW}(C, d) = \mathrm{SLT}(C, d) * \mathrm{SIDF}(C) \tag{7}$$

[14] See Definition 2.

[15] D is the entire documents-space.

[16] See ST at Definition 2.

[17] See C at Definition 3.

2.3 Semantic Distance Between Query and CS

As far as the semantic of concepts are considered, we need to accurately match each expanded word in the input query with each concept in CS, see Fig. 5. As stated before, not all matches of each semantic expansion have the same matching consistency.

For example, the first synonyms of the first query word, Qw_1s_1, may match one of the super-classes of the concept x in CS, C_xp_i. The Qw_1s_1 itself may matches one of the synonyms of another concept y, C_ys_j. Thus, as we justified earlier, the second match is stronger than the first. Therefore, for efficiency reasons, each case has to be handled separately. Otherwise, unexpected results will be produced. For instance, weak matches may take the same weights as other stronger ones. To that end, we generated the formulas (8) and (9) for constructing the entries of this sensitive-matching vector of the query.

Let n denotes the count of expansions that are matched between the query and the concepts in the space, and m_x denotes the frequency of each match. The Significance Level of the concept C in query Q is defined by Eq. 8. Where CAS_x is the Association Strength between the concept c and the match m_x, while the $QwAS_x$ is the Association Strength between the query word of the same mach.

Thus, the Semantic Weight of the concept C in the query Q is defined by Eq. 9. Accordingly, the Semantic Distance between query Q and document d_i is calculated

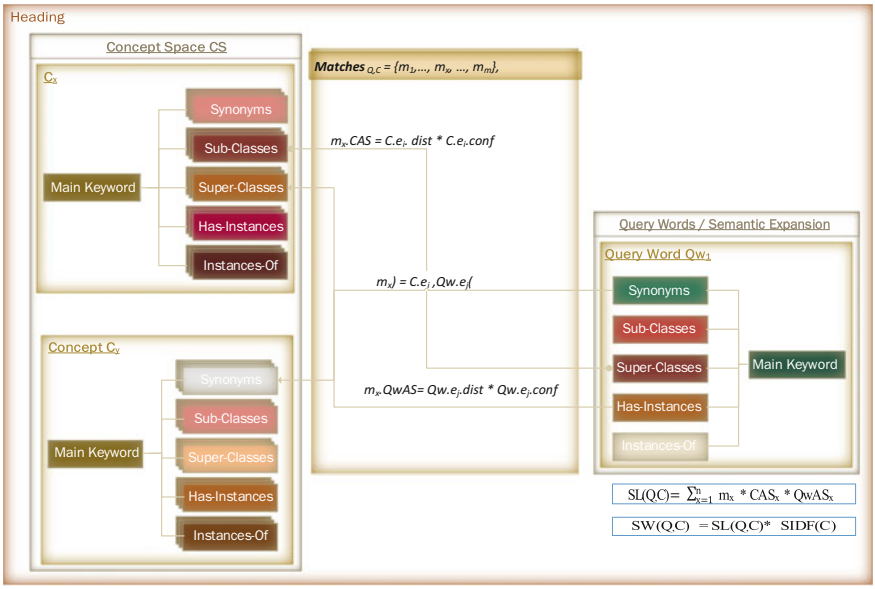

Fig. 5 Query word expansions and Concept's expansions matchings

regarding the semantic weights of the concept C_j in both the query Q and the document d_i by Eq. 10. Where q is the count of concepts in the space.

To sum up, we have described how to calculate weights of concepts in documents to construct the CS index and how to compare the concepts in an input query with these concepts to accurately determine relevant documents.

$$SL(Q, C) = \sum_{x=1}^{n} m_x * CAS_x * QwAS_x \tag{8}$$

$$SW(Q, C) = SL(Q, C) * SIDF(C) \tag{9}$$

$$SDist(d_i, Q) = \frac{\sum_{j=1}^{q} SW(Q, C) * SW(d_i, C)}{\sqrt{\sum_{j=1}^{q} SW(Q, C)^2} * \sqrt{\sum_{j=1}^{q} SW(d_i, C)^2}} \tag{10}$$

3 System Architecture

This section describes the architecture of the proposed system and its components. The overall architecture is portrayed in Fig. 6. Each process is presented in details at the phases of Fig. 7. The generation particulars of the Inflectional, Semantic, and Conceptual vectors are presented more formally by the Algorithms presented at the Appendix.

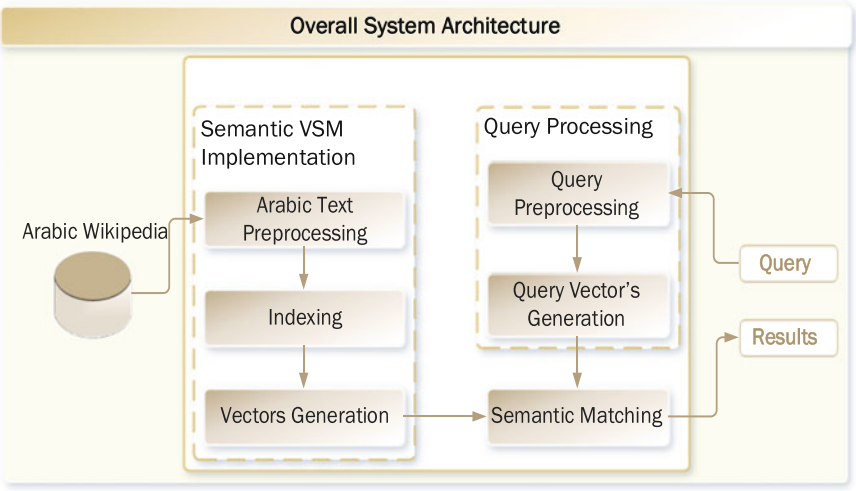

Fig. 6 Overall system architecture

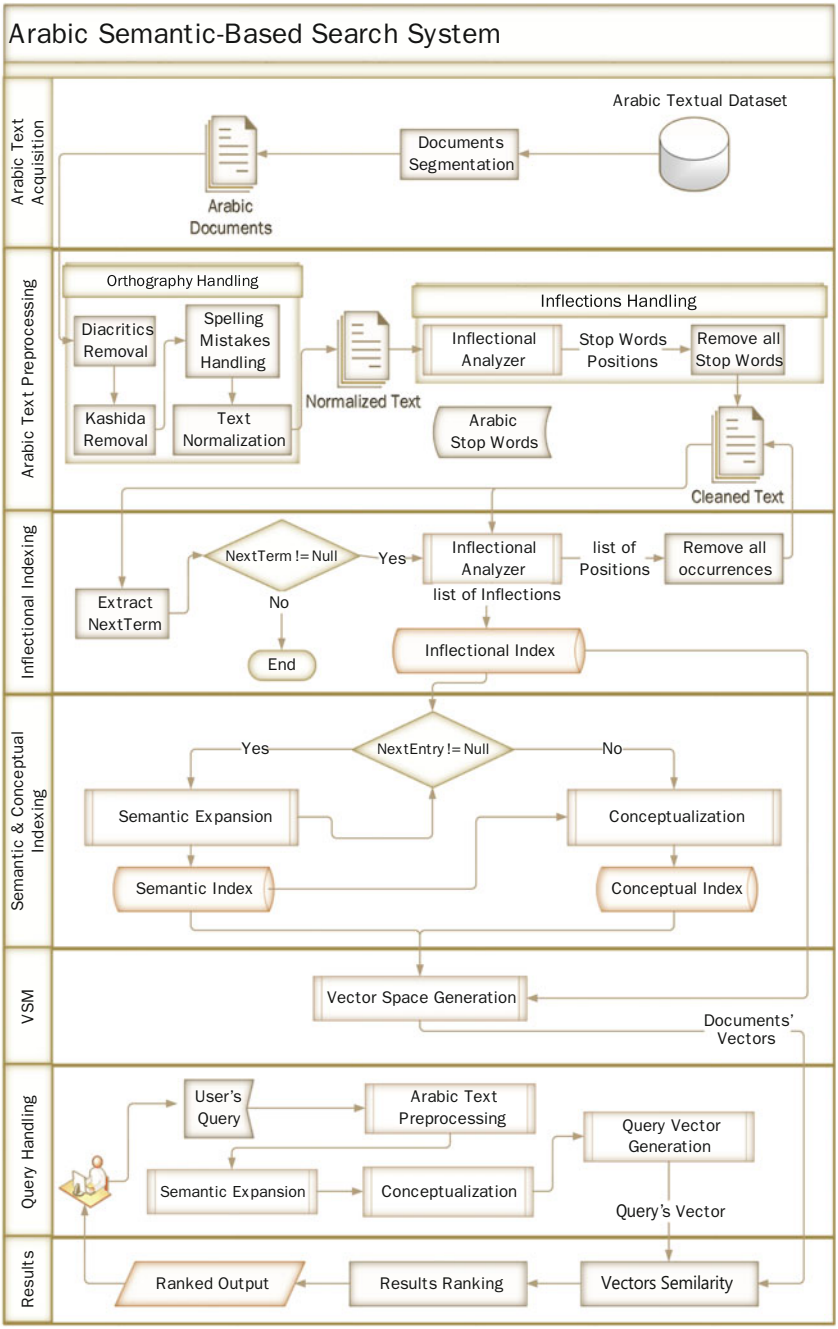

Fig. 7 Detailed system sub-processing

4 Experimental Analysis

4.1 Experimental Setup

The knowledge source is a set of documents extracted from a full dump of the Arabic Wikipedia.[18] We have conducted five experiments to test the effectiveness of the features proposed by the current research. The description of these experiments is presented in Table 1. The experimental results are divided into three interrelated dimensions: document representation, document retrieval, and document ranking. The objective is to measure the following (Fig. 8):

1-The indexing efficiency: when the conceptual indexing is adopted, the documents are retrieved according to their semantic concepts instead of the lexical terms. This dimension of evaluation aims to measure the efficiency of representing documents according to their central concept(s). As presented before, the weight of an index-entry[19] regarding to a document is referring to its capability of distinguishing this document in the space. Therefore, this evaluation is judged by the *Document Frequency (df)* and *Weight (W)* averages. The higher the *df* and *w* for an index-entry, the documents that are more relevant will be retrieved, which increasing the relevance accuracy. Besides the relevance accuracy, another dimension is needed for measuring the capability of the retrieval process itself.

2-The retrieval capability: the documents retrieval method does not only retrieve the documents that are syntactically or even semantically relevant, but also documents which are conceptually (ontologically) relevant. This evaluation aims to measure the impact of the indexing method on the performance of the retrieval process. It is judged by: Precision, Recall, and F-Measure.

However, the retrieved documents do not rank the same. The efficiency of the ranking process affects the overall performance of the search system. Besides measuring the accuracy of the retrieval process, another dimension is needed to measure the accuracy of the document ranking.

3-The accurate ranking: this evaluation aims to measure the accuracy of assigning the corresponding weight that exactly represents the association strength of each index entry with a document in the space. Therefore, the incidence indicator factor directly affects the capability of the accurate ranking. This evaluation is semi-automatic. It calculates the average of distance between the ranking order that results from each experiment and the ranking order judged by a human expert. The smaller the distance average, the closer the rank to the standard.

[18]The Arabic Wikipedia dump is accessed on 29-Aug-2012 from http://dumps.wikimedia.org/arwiki/.

[19]A term or hence a concept.

Table 1 The experiments description and the results summary

Experiment	Indexing	Expanding type	Index size	Incidence indicator	*tf* Average	*df* Average	*w* Average	Ranking-distance average	Experiment description
V1	TS	Inflectional	360486	IF	0.514	2.02	1.02	4.62	Inflectional frequency-based VSM
V2	STS	Semantic		SF	0.704	2.6	2.11	4.4	Semantic frequency-based VSM
V3				SLT			1.65	4.39	Significance-level-of-term based VSM
V4	CS	Conceptual	223502	CF	0.7076	3.8	2.83	4.165	Conceptual frequency-based VSM
V5				SLC			2.32	4.154	Significance-level-of concept based VSM

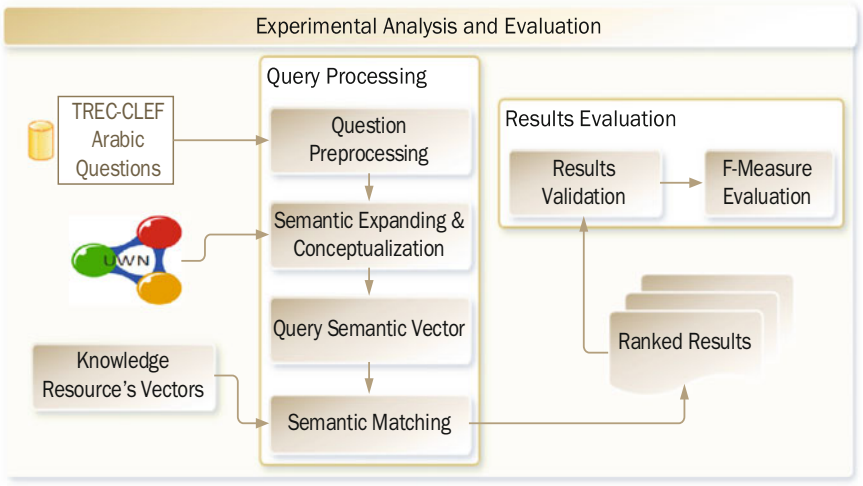

Fig. 8 Experimental analysis and evaluation

4.2 Experiments, Results, and Evaluation

To demonstrate the variations of the retrieval capability when each indexing method is applied, we measured their performance in terms of the values of *Document Frequency (df)*, *Term Frequency (tf),* and *Weight* (*w)*. Table 1 represents the summary of the results presented in the subsequent sections.

4.2.1 The Conceptualization Levels

From the *AWDS*, a terms-space (TS) of 360486 terms is extracted after excluding the Named Entities. The TS is then semantically expanded using the UWN to construct the STS as defined at Definition 2. The two conceptualization levels, presented at Algorithm 3, are then applied on the STS to generate the CS as presented at Definition 3. As a result, the size of the STS is shrunk by 38% to construct the CS version.

This leads to the increment of the representation power of each item in the space since the average of items weights is increased as shown Table 1. Note that the weight average of V3 and V5 are lower than those of V2 and V4, respectively. We are going to explain these results with the discussion of ranking accuracy. However, it is noticeable that the weights of the STS index are higher than those of the inflectional indexing TS. Moreover, the weights of the CS are greater than those of the STS. The important observation is that the obtained results are demonstrating the efficiency of the CS in distinguishing documents according to weights of the corresponding concepts.

4.2.2 The Retrieval Capability

The F-Measure is a very common measurement for calculating the performance of the retrieval systems. It is based on the harmonic mean of the Recall and the Precision scores, which we used to evaluate the retrieval accuracy of the proposed system. The F-Measure is defined by Eq. 11.

$$\mathrm{F-Measure} = 2 * \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{11}$$

4.2.3 The Ranking Accuracy

The ranking accuracy of the proposed system is evaluated by measuring the correctness of assigning the weight that precisely characterizes the Association Strength of each index with documents in the space. This is representing how the Incidence Indicator factor affects the ability of the accurate ranking.

The values of the weights averages of the experiments V2 and V4 are greater than those of V3 and V5 respectively. The ranking results show that the V3 experiment is better than V2, while V5 is the best. This is due to the extra error ratios caused by the use of the Sf and CF indicators instead of the SLT and SLC. However, these extra ratios are reduced as a result of using the SLT and SLC indicators at the V3 and V5 experiments, which is directly reflected on the ranking efficiency of these experiments. Still, the experiment V4 gives a better ranking order than V3, since it is based on the conceptual indexing CS, although it suffers from the extra error ratio caused by the CF indicator.

The ranking accuracy is calculated using the Distance Average (DA), Eq. 12, between the experimental order and the standard order delivered by a human specialist.

$$\mathrm{DA(V)} = \frac{\sum_{i=1}^{n} \left| \frac{1}{\mathrm{SRank_i}} - \frac{1}{\mathrm{ERank_i}} \right|}{n} \tag{12}$$

Where n is the count of the retrieved documents as a result of the user's query, while the $\mathrm{SRank_i}$ is the standard rank of document i, and the $\mathrm{ERank_i}$ is the rank is the experimental rank of document i at experiment V.

The closest ranking order is obtained at the experiment V5, which assures its parameters' capability to rank the retrieved result more accurately.

5 Conclusion and Future Work

This study sheds light on the inaptitude in searching the Arabic Language semantically, which may be attributed to the sophistication of the Arabic language itself. However, this should not stop more effective efforts for achieving the best possible solutions that enable the Arabic Language users getting the benefit from the new electronic technologies.

In an attempt to take a step in that long pathway, we proposed an Arabic semantic search system that based on the Vector Space Model. The VSM is one of the most common information retrieval models for textual documents due to its ability to represent documents in a computer interpretable form. However, as it is syntactically indexed, its sensitivity to keywords reduces its retrieval efficiency. To improve its effectiveness, the proposed system is extracting a concept-space index, using the universal wordnet UWN, to be used as a semantic index of VSM search system. The proposed system enables a conceptual representation of the document space, which in turn permits the semantic classification of them and thus obtaining the semantic search benefits. Moreover, we introduced a new incidence indicator to calculate the significance level of the concept in a document instead of the traditional term frequency. Furthermore, we introduced a new formula for calculating the semantic weight of the concept to be used in determining the semantic distance between two vectors. The system's experimental results showed an enhancement of the F-measure value using the conceptual indexing over that is based on the standard syntactic baseline.

As a future work, we have to solve some problems such as the ambiguity by discriminating the meaning contextually. Also, we may work on refining the processing of the multiword expression expansions. That will improve the results noticeably since. Moreover, the improvement of the Arabic knowledge representation in the UWN will help to overcome its limitations that directly affects the search results. Another open research area is to solve the problems of the Arabic language morphological analysis to prevent the consequent errors occurred in the indexing process, and hence, the construction of the search dictionary. We also may try to use Google Translation API with the UWN to find results for these terms that have results in languages other than Arabic.

Appendix: The Implementation Algorithms

Algorithm 1: V1

```
1. Input: D
2. Output: V1 and TS
3. Begin
4.   Indexing (D);
5.   Foreach dj in D
6.     term = GetNextTerm(dj)[20];
7.     VSM(term);
8.   EndFor
9.   Return V1, TS;
10. End
1.   Function VSM(t)
2.     TS.add(t);
3.     i = TS.size;
4.     Foreach dj in D
5.       inflections = Search (t, dj)[21];
6.       If inflections.size > 0
7.         TS.inflected_forms[i] = inflections;
8.         V1.df[i]++;
9.         V1.tf[i,j] = inflections.size;
10.      EndIf
11.    EndFor
12.    V1.IDF[i] = log(D.size / V1.df[i].size)
13.    Foreach dj in D
14.      V1.w[i,j] = V1.IDF[i] * V1.tf[i,j]
15.    EndFor
16. End VSM
```

Algorithm 2: V2 and V3

```
1.  Input: D and TS
2.  Output: V2, V3, and STS
3.  Begin
4.     STS.size = TS.size;
5.     STS.inflected_forms= TS.inflected_forms;
6.     Foreach Ti in TS
7.          STS.semantic_expansions[i] = UWN.Expansion(Ti); // see Def.2
8.     EndFor
```

[20]Getting the text term in *D* without redundancy.

[21]Search for any inflectional form of *t* in the document d_i using RDI Swift Searcher.

```
9.      Foreach ST_i in STS
10.       Foreach e_x in ST_i
11.         Foreach d_j in D
12.           semantic_inflections = Search(e_x, d_j)[22];

13.           If semantic_inflections.size > 0
14.             If j ∉ df[i]
15.                 V2.df[i].add(j);
16.                 V3.df[i] = V2.df[i];
17.             EndIf
18.             V2.tf[i,j]+= semantic_inflections.size;          //SLT(ST_i,d_j)
19.             AS = e_x.confidence * e_x.distance;              //AS(e)
20.             V3.tf[i,j]+= AS* semantic_inflections.size;      //SLT(ST_i,d_j)
21.           EndIf
22.         EndFor
23.       EndFor
24.       V2.IDF[i] = log(D.size / V2.df[i].size);               //SIDF(ST_i)
25.       V3.IDF[i] = log(D.size / V3.df[i].size);               //SIDF(ST_i)
26.       Foreach d_j in D
27.         V2.w[i,j] = V2.IDF[i] * V2.tf[i,j];                  //SW(ST_i,d_j)
28.         V3.w[i,j] = V3.IDF[i] * V3.tf[i,j];                  //SW(ST_i,d_j)
29.       EndFor
30.     EndFor
31.   Return V2,V3, STS;
32.  End
```

Algorithm 3: V4 and V5

```
1.   Input: V2, V3, and STS
2.   Output: V4, V5, and CS
3.   Begin

4.       Indexing(S)[23];
```

[22]Search for any inflectional occurrences for the semantic expansion e_x of the term ST in document d_j using RDI Swift Searcher.

[23]S is the set of all Synonyms of all terms in STS.

```
5.      //1st Conceptualization Phase
6.      group_ID = 0;
7.      Foreach STi in STS
8.          x = [STS. inflected_forms[i], STS.semantic_expansions[i].synonyms];
9.          s = (STS. inflected_forms. Except(STS. inflected_forms[i])) ∪
10.             (STS.semantic_expansions.synonyms.
                    Except(STS.semantic_expansions[i].synonyms));

11.     relatedTerms = Search(x, s)[24];

12.     If relatedTerms.size > 0
13.         group_ID++;
14.         Foreach r in relatedTerms
15.           G[group_ID].add(STS[r]);
16.         EndFor
17.     EndIf
18. EndFor

19. //2nd Conceptualization Phase
20. Foreach g in G
21.     concept = g;
22.     Foreach g' in G. Except(g)
23.         If g ∩ g' ≠ ∅
24.           concept ∪= g';
25.         EndIf
26.     EndFor
27.     CS.add(concept);
28. EndFor

29. //Update V2 and V3 to get V4 and V5.
30. Foreach Ci in CS
31.     Foreach STx in C
33.         V4.df[i] ∪= V2.df[STx];                    //SLC(Ci,dj)
34.         V4.tf[i,j]+= V2.tf[STx,j];                 //SLT(Ci,dj)
35.         V5.tf[i,j]+= V3.tf[STx,j];                 //SLT(Ci,dj)
32.     Endfor
33.     V5.df[i] = V4.df[i];
36.     V4.IDF[i] = V5.IDF[i] =  log(D.size / V4.df[i].size);    //SIDF(Ci)
37.     Foreach dj in D
38.         V4.w[i,j] = V4.IDF[i] * V4.tf[i,j];                  //SW(Ci,dj)
39.         V5.w[i,j] = V5.IDF[i] * V5.tf[i,j];                  //SW(Ci,dj)
40.     EndFor
34. Return V4,V5,CS;
35. End
```

[24]Search for any inflectional or Synonyms occurrences for the term ST_i in the set s using RDI Swift Searcher.

References

1. Shaalan, K.: A Survey of Arabic named entity recognition and classification (June 2014)
2. Saleh, L.M.B , Al-Khalifa, H.S.: AraTation: an Arabic semantic annotation tool (2009)
3. Tazit, N.: El Houssine Bouyakhf, Souad Sabri, Abdellah Yousfi. Semantic internet search engine with focus on Arabic language, Karim Bouzouba (2007)
4. Cardoso, J.: Semantic Web Services: Theory, Tools, and Applications. IGI Global, Mar 30 2007
5. Hepp, M., De Leenheer, P., de Moor, A.: Ontology Management: Semantic Web, Semantic Web Services, and Business Applications. Springer, New York; [London] (2008)
6. Kashyap, V., Bussler, C., Moran, M.: The Semantic Web: Semantics for Data and Services on the Web (Data-Centric Systems and Applications). Springer, 15 Aug 2008
7. Panigrahi, S., Biswas, S.: Next generation semantic web and its application, March 2011
8. Unni, M., Baskaran, K.: Overview of approaches to semantic web search, July–December 2011
9. Renteria-Agualimpia, W., López-Pellicer, F.J., Muro-Medrano, P.R., Nogueras-Iso, J., Zarazaga-Soria1, F.J.: Exploring the Advances in Semantic Search (2010)
10. Kassim, J.M., Rahmany, M.: Introduction to Semantic Search Engine. Selangor (2009)
11. Hahash, N.: Introduction to Arabic natural language processing. Association for Computational Linguistics, 30 August 2010
12. Jarrar, M.: Building a Formal Arabic Ontology (Invited Paper), Alecso, Arab League. Tunis, 26–28 April 2011
13. Al-Zoghby, A.M., Shaalan, K.: Conceptual search for Arabic web content. In: Computational Linguistics and Intelligent Text Processing—16th International Conference, vol. 9042, pp. 405–416, 14–20 April 2015
14. Al-Zoghby, A., Ahmed, A., Hamza, T.: Arabic semantic web applications: a survey. J. Emerg. Technol. Web Intell. 52–69 (2013)