Hindawi Publishing Corporation
Advances in Artificial Intelligence
Volume 2012, Article ID 620461, 8 pages
doi:10.1155/2012/620461

Research Article

Contribution to Semantic Analysis of Arabic Language

Anis Zouaghi,[1] Mounir Zrigui,[2] Georges Antoniadis,[3] and Laroussi Merhbene[2]

[1] *LATICE, ISSAT, University of Sousse, Cité Ibn Khaldoun, Taffala, 4003 Sousse, Tunisia*
[2] *LATICE, FSM, University of Monastir, Avenue de l'Environnement, 5000 Monastir, Tunisia*
[3] *LIDILEM, University of Stendhal, BP 25, 38040 Grenoble Cedex 9, France*

Correspondence should be addressed to Anis Zouaghi, anis.zouaghi@gmail.com

Received 11 June 2012; Revised 8 August 2012; Accepted 28 August 2012

Academic Editor: Srinivas Bangalore

We propose a new approach for determining the adequate sense of Arabic words. For that, we propose an algorithm based on information retrieval measures to identify the context of use that is the closest to the sentence containing the word to be disambiguated. The contexts of use represent a set of sentences that indicates a particular sense of the ambiguous word. These contexts are generated using the words that define the senses of the ambiguous words, the exact string-matching algorithm, and the corpus. We use the measures employed in the domain of information retrieval, Harman, Croft, and Okapi combined to the Lesk algorithm, to assign the correct sense of those proposed.

1. Introduction

Human language is ambiguous; many words can have more than one sense: this sense is dependent on the context of use. The word sense disambiguation (WSD) allows us to find the most appropriate sense of an ambiguous word. This work is a contribution in a general frame-work which aims at understanding the Arabic speech [1, 2]. In this paper, we are interested in determining the meaning of Arabic ambiguous words which we can meet in the messages transcribed by the module of speech recognition.

We propose some steps [3] to build a system for Arabic word sense disambiguation. First, we use a predefined list of stopwords (which do not affect the meaning of the ambiguous words) to eliminate them from the original sentence containing the ambiguous word. After that, we apply the routing [4] for the words contained in the glosses of the ambiguous word. Then we use the exact string-matching algorithm [5] to be able to extract the contexts of uses from the corpus used. Finally, we apply the measures of Harman [6], Croft [7], and Okapi [8] that compares the original sentence with the generated contexts of use and returns a score that corresponds to the closest context of use [9]. The Lesk algorithm [10] will be used to choose the exact sense from the different senses given by these measures.

This paper is structured as follows. We describe in Section 2 the main used approaches for WSD. After that, in Section 3, we present the proposed algorithm for lexical disambiguation of Arabic language. Finally, in Section 4, we present the obtained results.

2. Main Used Approaches

Most of the works related to the word sense disambiguation were applied to the English. They achieve a disambiguation rate of around 90%. There are many approaches which are classified using the source of knowledge adapted for the differentiation of the senses.

2.1. Knowledge-Based Methods. They were introduced in 1970, based on the dictionary, thesaurus, and lexicon. Using these resources they extract the information necessary to disambiguate words. Some of them [10] tested the adequate definitions given by the electronic dictionary Collins English Dictionary (CED) and the Dictionary of Contemporary English (LDOCE) for the automatic treatment of the disambiguation. Some others try to provide a basis for determining closeness in meaning among pairs of words described by the thesaurus like Roget International Thesaurus or by the semantic lexicon like Wordnet [11].

2.2. Corpus-Based Methods. Since the evolution of the statistic methods based on large text corpus, two principal orientations appear.

(i) Unsupervised methods: these methods are based on training sets and use a non-annotated corpus. They are divided into type-based discrimination [12] and token-based discrimination [13]: the first one used algorithms to measure the similarities after the representation of the contexts. The contexts are represented by high-dimensional spaces defined by word co-occurrences. The second one clusters the contexts that contain a specified target word such that the resulting clusters will be made up of contexts that use the target word in the same sense.

(ii) Supervised and semi-supervised methods: they use an annotated training corpus inducing the appropriate classification models [14]. For the supervised systems we can cite: the probabilistic methods; the majority of them use the naïve bayes algorithm and the maximum entropy approach. Methods are based on the similarity of the examples that use a similarity metric to compare the set of learned vector prototypes (for each word sense). The methods based on discriminating rules use selective rules associated with each word sense. The methods based on rule combine heterogeneous learning modules.

3. Proposed Method

As we have mentioned before, the majority of the works related to the WSD were applied to the English. However, there are some works applied to Arabic. We can state the unsupervised approach of Bootstrapping Arabic Sense Tagging [15], the naïve Bayes classifier for AWSD [16], the Arabic WSD by using the variants of Lesk algorithm [17], the WSD-AL system [18, 19], and so forth. Here, we define an unsupervised method named. Figure 1 below describes the principle of this method. We use the dictionary of "Al-Mu'jam Al-Wasit" to construct a database that contains the words and there definitions (an electronic version of this dictionary).

Subsequently we eliminate stopwords from the original sentence, using the list of stop words defined in our database (see Section 4.1.3). Using the glosses of the word to be disambiguated, we generate the contexts of use for each sense from the corpus. The idea consists of combining the algorithm of stemming (see Section 3.1.1) [4] to extract the roots and the algorithm of approximate string matching (see Section 3.1.2) to find occurrences of the stems. Stems and their occurrences are saved in the knowledge base. The sentences containing these occurrences with the ambiguous word represent the contexts of use.

The second step of the proposed method is to measure the similarity between the different contexts of use generated from the glosses and the current context. The context that obtains the highest score of similarity with the current context will represent the most probable sense of the ambiguous word. The Algorithm 1 below describes the proposed

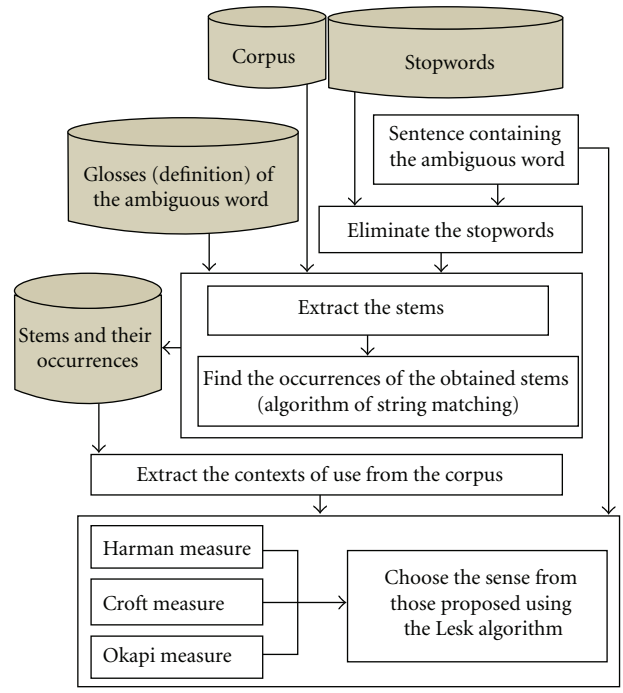

Figure 1: Principle of the proposed method.

algorithm of Arabic WSD and the score measure. In what follows we describe with more details each step cited above.

3.1. Construction of the Context of Use: Motivation and Implementation. To maximize the probability of finding the context for each gloss, we proposed as a solution to generate the occurrences of the most significant words. To extract the most significant words, we eliminate the non-informative words (stop words in English) using a predefined list (this list contains 20000 words). Given that the Arabic word has flexional morphology, we used an algorithm to extract the word roots and then an algorithm for matching words to find occurrences of this root. Obtaining instances of a root consists of adding a suffix to the beginning of a word or a prefix in the end.

3.1.1. Routing. To extract the stems of the Arabic words we use the Al-Shalabi-Kanaan algorithm [4]. Its advantage is that it does not use any resources. This algorithm extracts word roots by assigning weights to word's letters (the weights are real numbers predefined between 0 and 5) multiplied by the rank which depends on the letter's position. The roots are composed of three to five consonant letters and 85% of Arabic roots are trilateral. For that reason we use this algorithm which returns only three consonants.

The weights affiliated to letters were determined through experiments on Arabic texts, for example, we assign the most highest weight "5" to the letters "ا, ة" "a, t" because the words in the Arabic language begin and finish by these letters. The rank of letters in a word depends on the length of the word, and if the word contains an even or odd number of letters. The three letters with the lowest weights are selected.

S: Sentence contained the word to be disambiguated;
CU: Context of use generated;
C: Corpus; R_4: root;
G: Glosses from the dictionary generated in Section 3;
AW: ambiguous word;
w, y: word;
(1) For each $w \in S$ {
(2) Assign weight p = the position on the left or on the right of the ambiguous word;
}
(3) For each $w \in$ Glosses of AW {
(4) Lemmatizing (w); // Generate the root
(5) For each $y \in C$ {
(6) Approximate String-Matching (char w, int m, char y, int n); // Generate a list of occurrences $L(R_4)$;
}
(7) For each $w_1 \in L(R_4)$ {
(8) Load all the sentences that contains these occurrences to generate the context of use CU;
}}
Context-Matching (S, CU) {
(9) For each $w \in S$ {
(10) For each CU containing AW {
(11) $C(w) = -\log(n(w)/N) \times [0.5 + (1 - 0.5) \times (n_{\text{cu}}(w)/\text{Max}_{x \in \text{uc}}\ n_{\text{cu}}/(w))]$;
(12) $O(w) = \log[(N - n(w) + 0.5)/n(w) + 0.5] \times [n_c(w)/(n_{\text{cu}}(w) + (T(\text{cu})/T_n(B)))]$;
(13) $H(w) = -\log(n(w)/N) \times [\log(n_{\text{cu}}/(w) + 1)/\log(T(\text{cu}))]$;
(14) If the result given by each measure are different then
(15) Apply the leak algorithm between the proposed glosses proposed;
Else
(16) Affiliate the sense proposed by the different similarity measures;
}}

Algorithm 1: WSD-AL algorithm.

Table 1: Execution of the stemming algorithm to extract the root of the word "الحساب" "alhissab".

word	الحساب					
Letters	ب	ا	س	ح	ل	ا
Wheights	0	5	1	0	1	5
Rank	1.5	2.5	3.5	4	5	6
Multiplication	0	15	3.5	0	5	30
Root	حسب					

In Table 1, we give a sample of the execution of the algorithm for the word "الحساب" "alhissab".

This algorithm achieves accuracy in the average of 90% [10]. The output of this step is a list of the root of the words contained in the gloss of the ambiguous word $R(g_i) = \{R_1, R_2, \ldots, R_n\}$, where g_i is the ith gloss and R_n is the ith obtained stem.

3.1.2. Approximate String Matching. Unlike English, Arabic has a rich derivational system and it is one of the characteristics that makes it ambiguous. The Arabic words are based on roots, generally trilateral. We use the algorithm of approximate string matching [5] to find the possible occurrences of a stem obtained using the last step. This algorithm will be applied only for the roots that we do not find in our database.

It is based on two steps [20]. The first step (see Algorithm 2) consists of filling the matrix of the two words to be compared to x and t. Let $|x| < |t|$ and δ = substitution.

After that we use the step of back-tracking (see Algorithm 3), to find the shortest common subsequence. Let be γ the operation of insertion and $\sigma_{i,j}$ the operation of suppression. The words containing this common subsequence (stem obtained previously) will be considered as the occurrences of the stem. A list L(Ri) of the occurrences will be generated.

We use the corpus described in the experimental results, to extract the sentences containing the words of the glosses and their occurrences. These texts represent the contexts of use. This algorithm takes so much time during its execution; to facilitate that, we generated a table in our knowledge base in which are recorded occurrences of each root are recorded. Until now this table has a list of 7,349 roots with an average of seven occurrences for each root.

3.2. Score Measure: Motivation and Implementation. We propose some measure that determines the degree of similarity between a sentence (containing an ambiguous word) and a document (that represents the contexts of use for a given sense of the ambiguous word). Let CC = $m_1, m_2, \ldots, m_k$ the context where the ambiguous word m appear. Suppose that

```
Begin
(i) (i.a) Construct the matrix M with size (|x| + 1) ∗ (|t| + 1);    //
Filling the matrix
        (i.b) For i := 1 à |x| do M[i, 0] := i ∗ δ end;
                For j := 0 à |t| do M[0, j] := 0 end;
    (ii)        For i := 1 à |x| do
                  For j := 1 à |t| do
                      M[i, j]: = min{M[i − 1, j − 1] + 1,
                                     M[i, j − 1] + 1,
                                     M[i − 1, j] + δ}
                  End
                End
```

Algorithm 2: First step "filling the matrix" for the approximate string matching algorithm.

```
(iii) (iii.a) Select q, 1 ≤ q ≤ |t|
telle que M[|x|, q] = min_{1≤j≤|t|}{M[|x|, j]}; // Back-Tracking
                i := |x|; j := q;
        (iii.b) whiel (i ≠ 0 & j ≠ 0) do
    If M[i, j] = M[i, j − 1] + γ than j := j − 1
    else
                                if M[i, j] = M[i − 1, j − 1] + σ_{i,j} than
                                j := j − 1; i := i − 1
            else i := i − 1
                                end if
            end if
    end do;
    (iv) p := j + 1;
            x′ := t_{p,q}
end
```

Algorithm 3: Second step "back-tracking" of the approximate string matching algorithm.

$S_1, S_2, \ldots, S_k$ are the possible senses of m out of context. And $\mathrm{CU}_1, \mathrm{CU}_2, \ldots, \mathrm{CU}_K$ are the possible contexts of use of m for which the meanings of m are, respectively: $S_1, S_2, \ldots, S_k$.

To determine the appropriate sense of m in the current context CC we used the information retrieval methods (Okapi, Harman, and Croft), which allow the system to calculate the proximity between the current context (context of the ambiguous word), and the different use contexts of each possible sense of this word. The results of each comparison are a score indicating the degree of semantic similarity between the CC and given CU. This allows our system to infer the exact meaning of the ambiguous word. The following (1) describes the method used to calculate the score of similarity between two contexts:

$$\mathrm{S}_t(\mathrm{CC}, \mathrm{CU}) = \frac{(\Sigma_{i\in \mathrm{RC}} E(m_i) + \Sigma_{i\in \mathrm{LC}} E(m_i))}{(\Sigma_{i\in \mathrm{RC}} \mathrm{FE}(m_i) + \Sigma_{i\in \mathrm{LC}} \mathrm{FE}(m_i))}, \quad (1)$$

where $\Sigma_{i\in \mathrm{RC}} E(m_i)$ and $\Sigma_{i\in \mathrm{LC}} E(m_i)$ are, respectively, the sums of weights of all words belonging at the same time, the current context CC and the context of use (CU). $\mathrm{FE}(m_i)$ corresponds to the first member of $E(m_i)$, and $E(m_i)$ can be replaced by one of the information retrieval methods: Croft, Harman, or Okapi, whose equations are, respectively, as follows.

3.2.1. Harman Measure. Consider

$$H(m) = W_H(m, \mathrm{CU}(t)) = -\log\left(\frac{n(m)}{N}\right) \times \left[\frac{\log(n_{\mathrm{cu}}(m) + 1)}{\log(T(\mathrm{cu}))}\right], \quad (2)$$

where $W_H(m, \mathrm{CU}(t))$ is the weight attributed to m in the use contexts CU of the ambiguous word t by the Harman measure; $n(m)$ is the number of the use contexts of t containing the word m; N is the total number of the use contexts of t; $n_{\mathrm{cu}}(m)$ is the occurrence number of m in the use context CU; $T(\mathrm{cu})$ is the total number of words belonging to CU.

3.2.2. Croft Measure $C(m)$. Consider

$$C(m) = W_C(m, \mathrm{CU}(t)) = -\log\left(\frac{n(m)}{N}\right) \times \left[k + (1-k) \times \left(\frac{n_{\mathrm{cu}}(m)}{\mathrm{Max}_{x\in \mathrm{uc}} n_{\mathrm{cu}}(x)}\right)\right], \quad (3)$$

where $W_C(m, \mathrm{CU}(t))$ is the weight attributed to m in the context of use (CU) of t by the Croft measure; k is a constant

```
Begin
      Score ← 0
      Sens ← 1        // Choose the sense
      C ← context (t)        //Context of the word t
      For all I ∈ [1, N]
        D ← description (si)
      Sup ← 0
      For all w ∈ C do
          w ← description (w)
          sup ← sup + score (D, w)
      if sup > score then
        Score ← sup
        Sens ← i
End.
```

Algorithm 4: Simplified Lesk algorithm.

that determines the importance of the second member of $C(m)$ ($k = 0{,}5$); $\mathrm{Max}_{x\in \mathrm{uc}} n_{\mathrm{cu}}(x)$ is the maximal number of occurrences of word m in CU.

3.2.3. Okapi Measure. Consider

$$O(m) = W_O(m, \mathrm{CU}(t)) = \log\left[\frac{(N - n(m) + 0{,}5)}{n(m) + 0.5}\right] \times \left[\frac{n_c(m)}{(n_{\mathrm{cu}}(m) + (T(\mathrm{cu})/T_m(B)))}\right], \quad (4)$$

where $W_O(m, \mathrm{CU}(t))$ is the weight attributed to m in CU of t by the Okapi measure; $T_m(B)$ is the average of the collected use contexts lengths. This will enable us to increase the probability of finding the nearest context to the original sentence containing the ambiguous word.

3.2.4. The Simplified Lesk Algorithm. The Lesk algorithm, introduced in 1986, was derived and used in several studies of Pedersen and Bruce [21] and Sidorov and Gelbukh [22], and so forth. We can also cite the work of Vasilescu et al. [23] that evaluates variants of the Lesk approach for disambiguating words on the Senseval-2 English all words. This evaluation measures a 58% precision, using the simplified Lesk algorithm [24], and only a 42% under the original algorithm. The algorithm of Lesk is used to find the gloss that matches more with the candidate glosses of the words contained in the same sentence including the word to be disambiguated. This algorithm presented some limits (cited in paragraph 4.3) to generate the correct sense. Since that, we test a modified version of the Lesk algorithm using five measures of similarities. These measures will be applied to find the similarity between each sense of the ambiguous word proposed in AWN and the senses of the other words contained in the same sentence.

We adapted simplified Lesk algorithm [24] that adapts the Lesk algorithm [10] to calculate the number of words that appear in the current context of ambiguous word and the different contexts of use, which was considered as semantically closer to the results of methods used previously. The input of the algorithm is the word t and $S = (s_1, \ldots, s_N)$ are the candidate senses corresponding to the different contexts of use achieved by applying methods of information retrieval. The output is the index of s in the sense candidates. Algorithm 4 below details the simplified Lesk algotithm.

The choice of the description and context varies for each word tested by this algorithm.

The function context (t) is obtained by the application of the input context. The function description (si) finds all the candidate senses obtained by the information retrieval methods. The function score returns the index of the candidate sense: score (D, w) = Score (description $(s), w)$.

The application of this algorithm allowed us to obtain a rate of disambiguation up to 76%.

4. Experimental Results

To check the validity of the algorithm presented in the previous section, tests were conducted using some free tools. The English works were evaluated using Senseval-1 or Senseval-2. However in our work we have to make our experimental data using a totally different set of resources. To measure the rate of disambiguation, we use the most common evaluation techniques, which select a small sample of words and compare the results of the system with a human judge. We use the metric of the precision P (see (5)), recall R (see (6)), and finally the balanced F-score which determines the weighted harmonic mean of precision and recall (see (7)):

$$P = \frac{\text{correct answers provided}}{\text{answers provided}}, \quad (5)$$

$$R = \frac{\text{correct answers provided}}{\text{total answers provided}}, \quad (6)$$

$$F\text{-score} = \frac{2(P \times R)}{(P + R)}. \quad (7)$$

After that, as an upper bound, we use the context of use that corresponds to the most frequent sense (MFS).

Table 2: Description of the used dictionary.

Number of letters	Number of pages	Average number of glosses per word
29	1407	12 glosses/words

Table 3: Characteristics of the collected corpus.

Measure	Value
Total size of the corpus	1500 texts
Number of ambiguous words	50 words
Average number of synonyms of each ambiguous word	4
Average number of the possible senses	12
Average size of each context of use	970 words, 130 sentences
Average size of the text	500 words

4.1. Used Tools and Experimental Data

4.1.1. Dictionary. We use the dictionary of "Al-Mu'jam Al-Wasit" that contains the Arabic lexicography. Therefore, we construct a database that contains the words of an electronic version of this dictionary and their glosses. Table 2 below describes the characteristics of the dictionary.

We give in what follows a sample of glosses for the word "عين" "ayn" given by the dictionary Al-Wasit.

First gloss

عضو الإبصار للإنسان وغيره من الحيوان

Transcription

Organ vision of man and other animals.

Second gloss

يَنْبُوعُ الماء ينبُعُ من الأرض ويجري

Transcription

Fountain water flows from the land being.

In this work we choose to work on fine-grained senses. This choice makes our work more difficult and complex because it increases the number of the considered senses.

4.1.2. Corpus. We chose to work on texts dealing with multiple domains (sport, politics, religion, science, etc.). These texts are extracted from newspaper articles, which were recorded in the corpus of Al-Sulaiti and Atwell [25]. Table 3 below describes the characteristics of the collected corpus.

These documents have the advantage of possessing an explicit structure that facilitates their presentation and their exploitation in different contexts to find relevant words more efficiently.

4.1.3. Stopwords. We have compiled a list of stop words which have no influence on the meaning of the sentence. This list contains 20000 empty words or stop words. To build this list, we collected from the net pronouns, noun, names, letters, noun-verb, and some words considered insignificant by humans.

Table 4: Results obtained by different measures after and before pretreatment.

Method	Without rooting	Without string-matching	Final rate	MFS
P	0.52	0.61	0.78	0.86
R	0.39	0.52	0.65	0.74
F-score	0.44	0.56	0.71	0.84

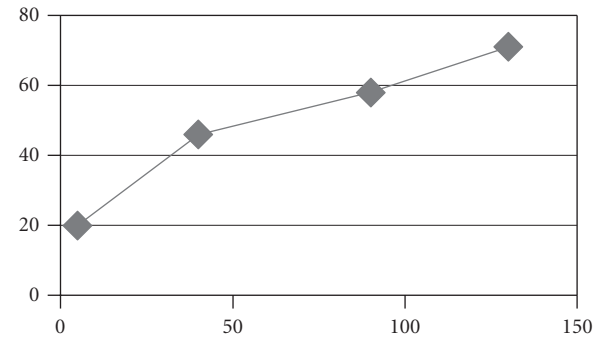

Figure 2: The F-Score obtained by varying the size of the context of use.

4.1.4. Experimental Data. Fifty words have been chosen. For each one of these ambiguous words, we evaluate 20 examples per sense. This number may be judged as not enough due to the problems encountered during the experimentation cited in what follows.

(i) The important number of glosses given by a dictionary for the ambiguous word.

(ii) The problem of the sentence segmentation due to the ambiguity of the Arabic language [1].

(iii) Finding the samples for the tests that can be judged as well as not so different for the process of disambiguation.

4.2. Obtained Results

4.2.1. Influence of the Stemming and String-Matching. We measure the performance of our system using the metrics presented above, with and without the respective use of the stemming algorithm and the string-matching algorithm (see Table 4).

Table 4 shows that the combination of the stemming algorithm with the string-matching algorithm gives the best results.

4.2.2. Influence of the Size of the Use Contexts. To determine the size of the collected context of use for each sense, we evaluate the results given by our system varying the size of that context of use (50 words, 100 words, and 150 words).

Figure 2 shows how the performance varies across the size of the context of use. We conclude that the lowest rate of disambiguation is mainly due to the insufficient number of contexts of use, which results in the failure to meet all

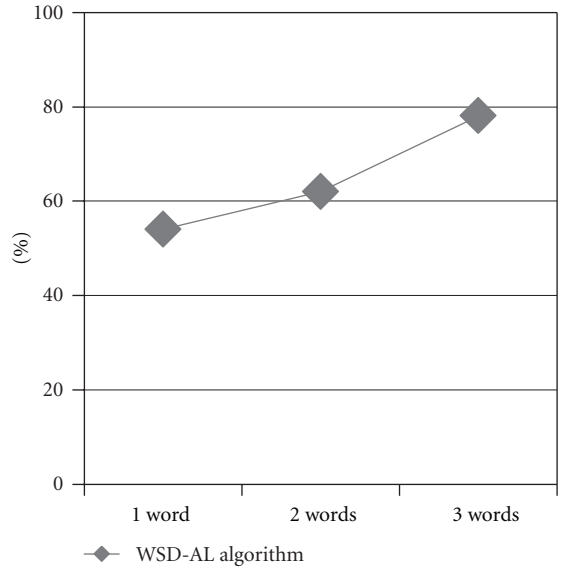

Figure 3: Results obtained for different window sizes.

possible events. For that we try to collect as many texts as we can, to extend the size of the knowledge database.

4.2.3. Influence of the Window Size. In the work of Yarowsky [26], a study of the influence of the window size on WSD shows that the most useful keywords for the WSD are included in a micro-context from six to eight words. However, we have to point out that in a so large context, it is difficult to discern the key elements for determining the meaning of a word. It seems obvious that a fixed size of the context window is not adapted for all the words. In order to solve this problem, we suggest determining the optimal size of the appropriate context for each test. We use a window size of 3 words (three words on the left and three words on the right of the ambiguous word), 2 words and one word. Figure 3 below shows as a final result of the experience the fact that the best rate of precision (P) and recall (R) is obtained for the word عين (ayn), especially by using a window size of three words.

The best similarity measure is obtained using a window size of three words. The Croft measure was the best one between those proposed.

4.2.4. Comparison of the Similarity Measures. Figure 4 presents a comparison between the results given by the Lesk algorithm and those given by the Croft, Harman, and Okapi measures. These results are shown for ten of the fifty words evaluated. This figure shows that the Lesk algorithm ameliorates the rate of disambiguation.

5. Conclusion

This paper has presented an unsupervised method to perform word sense disambiguation in Arabic. This algorithm is based on segmentation elimination of stop words, stemming

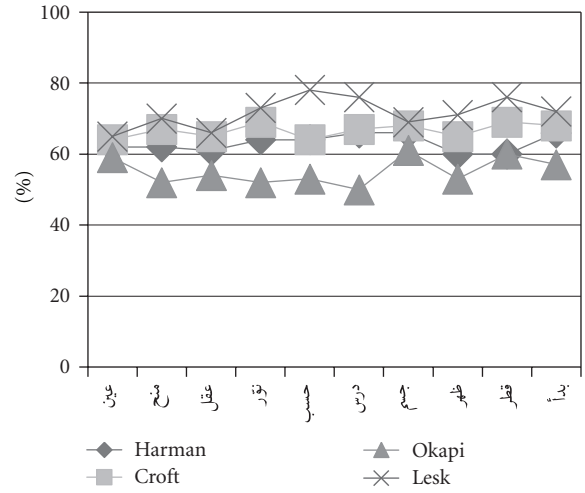

Figure 4: Comparison of the similarity measures.

and applying the approximate string matching algorithm for the words of the glosses. We measure the similarity between the contexts of use corresponding to the glosses of the word to be disambiguated and the original sentence. This algorithm will affiliate a score for the most relevant sense of the ambiguous word. For a sample of fifty ambiguous Arabic words that are chosen by their number of senses out of context (the most ambiguous words), the proposed algorithm achieved a precision of 78% and recall of 65%.

We propose that in future works, we ameliorate the correspondence between words and their glosses to build a system based on rules to disambiguate Arabic words.

References

[1] A. Zouaghi, M. Zrigui, and G. Antoniadis, "Automatic understanding of Spontaneous arabic speech—a numerical model," *TAL*, vol. 49, no. 1, pp. 141–166, 2008.

[2] M. Belgacem, A. Zouaghi, M. Zrigui, and G. Antoniadis, *Amelioration of the Performance of a Semantic Analyzer for the Comprehension of the Spontaneous Arabic Speech*, Applied Imagery Pattern Recognition, Washington, DC, USA, 2009.

[3] A. Zouaghi, L. Merhbene, and M. Zrigui, "A hybrid approach for arabic word sense disambiguation," *IJCPOL*. In press.

[4] H. Al-Serhan, G. Kanaan, and R. Al-Shalabi, "New approach for extracting Arabic roots," in *Proceedings of the Arab conference on Information Technology (ACIT'2003)*, pp. 42–59, Alexandria, Egypt, 2003.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, chapter 34, MIT Press, Cambridge, Mass, USA, 1990.

[6] D. Harman, "An experimental study of factors important in document ranking," in *Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 186–193, 1986.

[7] W. Croft, "Experiments with representation in a document retrieval system," *Research and Develpment*, vol. 2, no. 1, pp. 1–21, 1983.

[8] S. Robertson, M. Walker, and M. Gatford, Okapi at TREC-3, TREC-3, NIST Special Publication, 1994.

[9] A. Zouaghi, L. Merhbene, and M. Zrigui, "Combination of information retrievalmethods with LESK algorithm for arabic

word sense disambiguation," *Artificial Intelligence Review*, vol. 38, no. 4, pp. 257–269, 2012.

[10] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," in *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC'86)*, pp. 24–26, 1986.

[11] S. Banerjee and T. Pedersen, *Adapting the lesk algorithm for word sense disambiguation to WordNet [M.S. thesis]*, Partial Fulfillment of the Requirements, 2002.

[12] S. Derwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshmann, "Indexing by latent semantic analysis," *Journal of the American Society for Informartion Science*, vol. 41, pp. 391–407, 1990.

[13] T. Pedersen and R. Bruce, "Distinguishing word senses in untagged text," in *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, 1997.

[14] E. Agirre and P. Edmond, *Word Sense Disambiguation: Algorithms and Applications*, Springer, New York, NY, USA, 2006.

[15] M. Diab and P. Resnik, "An unsupervised method for word sense tagging using parallel corpora," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, pp. 255–262, Association for Computational Linguistics, Philadelphia, Pa, USA, 2002.

[16] S. Elmougy, H. Taher, and H. Noaman, "Naïve bayes classifier for arabic word sense disambiguation," in *Proceedings of the Neuro-Linguistic Programming (NLP)*, 2008.

[17] A. Zouaghi, L. Merhbene, and M. Zrigui, "Word sense disambiguation for arabic language using the variants of the lesk algorithm," in *Proceedings of the International Conference on Artificial Intelligence (ICAI'11)*, vol. 2, pp. 561–567, 2011.

[18] L. Merhbene, A. Zouaghi, and M. Zrigui, "Ambiguous Arabic words disambiguation," in *Proceedings of the 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD'10)*, pp. 157–164, London, UK, June 2010.

[19] L. Merhbene, A. Zouaghi, and M. Zrigui, "Arabic word sense disambiguation," in *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence (ICAART'10)*, pp. 652–655, January 2010.

[20] M. Elloumi, "Comparison of strings belonging to the same family," *Information Sciences*, vol. 111, no. 1–4, pp. 49–63, 1998.

[21] T. Pedersen and R. Bruce, "Distinguishing word senses in untagged text," in *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, 1997.

[22] G. Sidorov and A. Gelbukh, "Word sense disambiguation in a spanish explanatory dictionary," in *Proceedings of the Tratamiento Automático de Lengauje Natural (TALN'01)*, pp. 398–402, Tours, France, 2001.

[23] F. Vasilescu, P. Langlais, and J. Lapalme, "Evaluating variants of the lesk approach for disambiguating words," in *Proceedings of the Language Resources and Evaluation*, pp. 633–636, Lisbon, Portugal, 2004.

[24] F. Vasilescu, *Monolingual corpus disambiguation by the approaches of lesk [M.S. thesis]*, University of Montreal, Faculty of Arts and Sciences, Faculty of Graduate Studies, 2003.

[25] L. Al-Sulaiti and E. Atwell, "The design of a corpus of contemporary arabic," *International Journal of Corpus Linguistics*, vol. 11, no. 2, pp. 135–171, 2006.

[26] D. Yarowsky, "One sense per collocation," in *Proceedings of the ARPA Workshop on Human Language Technology*, pp. 266–267, Princeton, NJ, USA, 1993.