Lexical Disambiguation of Arabic Language: An Experimental Study

Laroussi Merhben, Anis Zouaghi, and Mounir Zrigui

***Abstract*—In this paper we test some supervised algorithms that most of the existing related works of word sense disambiguation have cited. Due to the lack of linguistic data for the Arabic language, we work on non-annotated corpus and with the help of four annotators; we were able to annotate the different samples containing the ambiguous words. Since that, we test the Naïve Bayes algorithm, the decision lists and the exemplar based algorithm. During the experimental study, we test the influence of the window size on the disambiguation quality, the derivation and the technique of smoothing for the (2n+1)-grams. For these tests the exemplar based algorithm achieves the best rate of precision.**

***Index Terms*—Supervised algorithms, training data, Naïve Bayes, decision list, exemplar based algorithm, window size.**

I. Introduction

HUMAN language is ambiguous; many words can have more than one sense that is dependent on the context of use. The word sense disambiguation (WSD) allows us to find the most appropriate sense of the ambiguous word.

The benefits of WSD were exploited by many NLP applications such as machine translation, information retrieval, grammatical analysis, speech processing as well as text processing.

The task of identifying the correct sense for the ambiguous word is not simple as it appears. What should be done to disambiguate a word? We must find a way to define the possible meanings of the word, since that we have to assign each occurrence of the ambiguous word to the appropriate sense.

In this work, we use the Naïve Bayes method, the decision lists and the exemplar based algorithm. These methods are based on a training phase during this part of work, we use an annotated training corpus (we extract from a non-annotated corpus the different samples containing the ambiguous word and we tag them with their senses).

Since that, a testing phase will classify a word into senses [1, 2]. In the most WSD works that was evaluated in the conference Semeval 2007, the supervised methods achieve the best disambiguation quality (about 80% precision and recall for coarse-grained WSD).

The paper is structured as follows. We describe in Section II how we tag the samples. After that, in Section III, we give a detailed account of the used supervised methods applied for the Arabic language. In Section IV, we present the results and discuss the difference with some related works in Section V. Finally, Section VI concludes the paper.

Manuscript received June 18, 2012. Manuscript accepted for publication July 24, 2012.

Laroussi Merhben Anis Zouaghi, and Mounir Zrigui are with the Unité de Recherche en Technologies de l'Information et de la Communication of the Réseau National Universitaire Tunisien, Tunisia (e-mail: aroussi_merhben@hotmail.com; Anis.Zouaghi@gmail.com; mounir.zrigui@fsm.rnu.tn).

II. Related works

A. Review Stage

We can cite the work of Mona Diab that uses a supervised learning approach called "bootstrap" [15]. This approach is highly accurate in the average of 90% of the evaluated data items based on Arabic native judgment ratings and annotations. Also, we find the work of Elmougy [16], where the Naïve Bayes algorithm was applied for the Arabic language. Some pre-treatement steps were applied like word rooting and eliminating stopwords, since that they use the net and a dictionary to collect ten training samples to each word for the testing phase. This work achieves a rate of precision of 73%. Compared to our work the amount of data is more less, and collecting the testing samples from the net is a hard task and not sufficient.

Finally, Soha M. Eid [17] compared the Rocchio Classifier to Naïve Bayesian classifier, the most frequent sense and the support vector machine using arabic lexical samples. The Rocchio classifier achieves an overall accuracy of 88% as the best rate and reduces the error by over 14%. But they test only five ambiguous words and they haven't explained how tagging the samples of the training phase.

Compared to our work we obtain a less rate of precision because of the important number of ambiguous tested words (fifteen ambiguous words). Also as a comparative study there is no test for the influence of the window size, the stemming and the smoothing on the quality of disambiguation.

For the other English related works, we can cite the experimental study that com-pares some supervised algorithms to disambiguate six senses of the word line [18] and [19]. Also the work of Pedersen where he compared the Naïve Bayes with Decision tree, Rule based learner, etc., to disambiguate the word line and 12 other words [20]. All these works, found that the Naïve Bayes algorithm performed as well as the other supervised algorithms, which is the same results founded in this work. Compared to the number of tested words by the English related works, we have to point that we test fifteen words for a derivational language that suffers from the lack of resources.

We can also compare the obtained results by some works of unsupervised Arabic word disambiguation, where the same samples and the same words were tested. In the first work

[21], it was proposed to use some information retrieval measures with the Lesk algorithm and it achieves a rate of 73%. In the second one [22], a Context matching algorithm returns a semantic coherence score corresponding to the context of use that is semantically closest to the original sentence. This algorithm achieves a precision of 78%. In this work, we obtain a less rate of precision. We can presume that the supervised works are more satisfactory for the task of Arabic Word Sense Disambiguation.

III. Methodology

This study experiments some supervised methods for the Arabic Word Sense Disambiguation. It compares the use of the Naïve Bayes algorithm, the decision list and the k nearest neighbor. These methods were used previously in many related works (that will be discussed in the section 5).

We have applied some pre-processing steps to the words belonging to the original sentence and the training sets.

A. Pre-processing

1) Extraction of stopwords

Over the past ten years several methods have been proposed for the extraction of stopwords that have no influence on the meaning of a given word [5], [6] and [7]. These methods are used to evaluate the significance of a word in a document, which also varies depending on the frequency of the word in the corpus. Thus allowing us to eliminate the stopwords (words that have no influence on the meaning of the ambiguous word) such as (حتى, من, قد, بها, في, كان, له, فوق) (even, of, may, by, in, was, to him, over the).

These words will be removed from the sentence containing the ambiguous word, to decrease the number of compared words.

In this part of our work, we use the tf-idf metric [8] that use the term frequency (tf) and the inverse document frequency (idf) (see equation 1).

$$\text{Tf-idf}_{i,j} = \text{tf}_{i,j} \times \text{idf}_i \quad (1)$$

The frequency of a word (Term Frequency) is the number of occurrences of a word in a given document. Let the document dj and the word ti, the frequency of ti in dj is measured as follows (see equation 2).

$$\text{tf}_{i,j} = n_{i,j} / \sum_k n_{k,j} \quad (2)$$

Where $n_{i,j}$ is the number of occurrences of the word w_i in the document d_j. The denominator is the number of occurrences of all words in the document d_j.

The inverse document frequency gives the importance of a word in the corpus. It's measured as follows (see equation 3).

$$\text{idf}_i = log \frac{|D|}{|\{dj : ti \in dj\}|} \quad (3)$$

where | D | is the total number of documents in the corpus and $|\{d_j : t_i \in d_j\}|$ is the number of documents where the word t_i appears. We have to note, that the elimination of stopwords, will decrease the number of compared words in the testing phase.

2) Stemming

Each Arabic word, nouns or verbs, is based on three letters, or more rarely four or two letters. These three letters are the root of the Arabic word, they are the most important letters used to be compared with other letters used for the derivation of the word (added to the right or left of the root).

In this work, we use the Khoja stemmer that removes the longest suffix and the longest prefix. It then combines the remaining words with verbal and nominal patterns, to extract the root [9].

The stemming were applied to the words contained in the original sentence, to find there occurrences in the extracted training samples.

B. Tagging Samples

The supervised methods need a training phase that used a tagged corpus. The examples obtained by the training phase, must contain as many words surrounding the ambiguous words as it will be needed in the test phase. We chose to work on texts dealing with multiple domains (sport, politics, religion, science, etc.). These texts were recorded in the corpus of Latifa Al-Sulaiti [3].

Using this corpus, we tag the founded ambiguous words (used in the testing phase) by their senses, this step was achieved with the help of four annotators (Arabic language teachers), that choose the ambiguous words by the important number of senses out of context. Using the dictionary Lissan al arab [4] which is one of the most famous Arabic dictionary, we were able to tag the words with their corresponding senses.

We haven't found an important difference between the sense tags, the arrangement between the annotators is in the average of 95%. In table 1, we give the statistics of the extracted samples.

TABLE I
DISTRIBUTION OF THE SENSES IN THE EXTRACTED SAMPLES.

Avg. # of words per sentence	Avg. # of senses per word	Avg. # of senses per ambiguous words	Avg. dominant sense for the ambiguous word
9,42	1,56	6,32	74%

Fifty words have been chosen. For each one of these ambiguous words, we evaluate 20 examples per sense. This number of words is judged as sufficient compared to the Senseval evaluation that put into practice 15 nouns, 13 verbs, 8 adjectives and 5 words that the grammatical tags wasn't taken into consideration. Totally there are 206 tests for every word.

C. Supervised methods

During the training phase, we tag the words surrounding the ambiguous word with $C_{i,j}$ (which is the local collocation that will indicate the position of two words given the ambiguous word). Let the ambiguous sentence is: " الفرار من الواقع إلى العلم والكتب هو سبيل اللقاء بين هذين العالمين المختلفين" "Escape from reality to science and books is the way of the meeting between these two different worlds".

Let "الكتب" "books" is the ambiguous word, in this sentence we can find 156 collocations. For example the collocation $C_{-2,-4}$: "الواقع_العلم" "fact _science", and $C_{-2,3}$: "العلم_ اللقاء" "science_ meeting".

For the original sentence, we define m features that correspond to the m collocations surrounding the ambiguous words. The supervised methods cited in what follows, will use these features for each sense of the ambiguous word.

1) The Naïve Bayes rithmalg

This method is one of the most popular and performant probabilistic method [10], it was used in different works of natural language processing including the word sense disambiguation. In fact the comparative works of Perdersen [11] found that the Naïve Bayes gives sufficient results compared to the other methods.

After tagging the samples containing the ambiguous word (AW) (section 3), we have to measure the probability of the collocations (F_j) contained in the same context of use of the ambiguous word for the sense (S) (see equation 4).

$$P = \sum_{j=1}^{m} \frac{\text{Number of occurrences of } F_j \text{ with the sense } S_i}{\text{number of occurrences of } F_j} \quad (4)$$

Where; N is the number of collocations in the original sentence.

This step is followed by the measure of the probability for each sense in the corpus (see equation5).

$$P(S_i) = \sum_{i=1}^{k} \frac{\text{Number of occurrences of AW with the sense Si}}{\text{Total number of occurrences of the ambiguous word}} \quad (5)$$

For the different collocations contained in the original sentence, we add the logarithm of the probability. The score is the sum of the obtained results (see equation 6).

$$\text{Score } (S_i) = \text{argmax}_{S_i \in senses(w)} \; log\, P(S_i) \prod_{j=1}^{m} P(F_j|S_i) \quad (6)$$

The sense with the highest score is the correct sense.

2) The decision List

The decision list algorithm was adopted for the WSD, in the work of Yarowsky [12]. In this part of our work, we need to compute the conditional probability of each sense for every collocation contained in the local context, p (Si | F1,F2,…, Fm) (see equation 7), for that we use :

- The probability of each observed collocation given the sense of the ambiguous word p (F_1,F_2,…, F_m |S_i) (see equation 1),
- The probability of each sense in the corpus p(S_i) (see equation 5) ,
- The unconditional probability of features (collocations surrounding the ambiguous word) (see equation 7).

$$P = p(F_1,F_2,\ldots,F_m|S_i) \times p(S_i) \,/\, p(F_1,F_2,\ldots,F_m) \quad (7)$$

Since that to construct the decision list, we have to sort the different obtained results (given by the equation 7 for the different senses of the ambiguous word) using the log of the conditional probability of two compared senses for the tested word (see equation 8).

Finally the surrounding collocations that obtain the highest score, will be attributed to that sense, and will be ranked in the top of the decision list. After this step of classification, we will obtain an ordered list of S_i given the obtained score.

$$\text{Score } (w_i) = \text{Abs } (\log (P(S_1|w_i) / P(S_2|w_i))) \quad (8)$$

Given the score obtained in the decision list, we can judge the significance of the words contained in the original sentence.

3) *The exemplar based (K Nearest Neighbor) algorithm*

The k nearest neighbor algorithm (KNN) is one of the highest performing methods in WSD [2]; [8]. The KNN algorithm is based on the k nearest similar instances to the tested instance. The classification phase is achieved by measuring the distance between the new example x = (F_1, F_2,…, F_m), and the previously stored examples x_i = (F_{i1} , . . . , F_{im}), to do that in this work we use the hamming distance (see equation 9).

$$\Delta(x, x_i) = \sum_{j=1}^{m} d_j \, \delta(F_j, F_{ij}) \quad (9)$$

Where (d_j = Number of occurrences of the j^{th} collocation in the previously stored examples / Total number of collocations) and $\delta(F_j, F_{ij})$ is 0 if F_j = F_{ij} and 1 otherwise. Since that, we can establish the set of the k most nearest examples. The most frequent sense between those k obtained samples is considered as the correct sense.

IV. Experimental Results

A. Encountered problems

Many problems have been encountered during the process of disambiguation cited in what follows:

- For the Naïve Bayes algorithm, we have the problem of the zero counts. As a solution, we replace the zero with $P(S_k)/N$, where N is the total size of the training sets. This solution is called smoothing.
- The important number of glosses given by a dictionary for the ambiguous word and the difference between the founded senses in the corpus. In the table 2, we give the number of senses for some words and their corresponding founded senses in the training corpus. As a solution, we try to collect from the net some texts containing the missing senses and add them to our corpus.
- Finding the samples for the tests (that can be judged effective and adequate for the process of disambiguation) is a hard task and differs between works for the obtained results.
- For some considered words, we have found senses that appear in the corpus and don't exist in the dictionary. These senses were added to the list of candidate senses. For the word "ayn" we extract about ten sentences from the training corpus where it means a name of a city in United Arab Emirates. A sample is given in what follows:

"تستقبلنا مدينة العين ببهاء يختلف تماما عما ألفناه في أبوظبي" → "The ayn city receives us brightly completely different than abu-dhabi".

– The difference between the number of occurrences in the corpus for each sense. For example for the word "كتب" "kataba", we have found about 452 samples where the considered sense is "write" which is the most frequent sense and about 23 samples where the considered sense is "predestined".

TABLE II
DIFFERENCE OF SENSES BETWEEN THE DICTIONARY AND THE EXTRACTED SAMPLES.

Ambiguous words	transcription	Number of senses in the dictionary	Number of senses in the extracted samples
حسب	hassaba	15	6
كتب	kataba	8	5
عين	ayn	20	8
شعر	chaar	8	4
عقل	aakl	18	6

1) Obtained results

To test the effectiveness and the impact of the different methods (presented in the previous section 4) on Arabic word sense disambiguation, we performed some experiments. In the work of Yarowsky [14], 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. Tests were conducted by measuring the performance (Precision) of each method varying the window size (the tested sentence containing the ambiguous word).

In Table 3, we give the rate of precision (Correct answers obtained / Obtained results) obtained using the bigram ($n = 2$), where we test only one word after or before the ambiguous word. In the case of trigram ($n = 3$) two words will be considered (one after the ambiguous word and another one before). Finally in the $(2n + 1)$-grams, we take into consideration n words surrounding the target word, in this experiments $n = 3$, because it give us the better results.

During the experimentation phase, we change the number of samples taken into consideration from the training phase. For example the 25% of samples means that we have taken into account 25% from the total number of samples that we have obtained in the training phase. The Figure 1 shows how the rate of precision varies across the percentage of samples. We conclude that the lowest rate of disambiguation is mainly due to the insufficient number of samples, which result in the failure to meet all possible events. For that we try to collect as many texts as we can, to extend the number of samples.

TABLE III
RESULTS OBTAINED BY DIFFERENT METHODS VARYING THE WINDOW SIZE.

Methods / tests	Naïve Bayes		Decision List		KNN	
	P	MFS	P	MFS	P	MFS
Bigram	23.04	29.17	21.43	25.29	26.33	30.21
Trigram	34.19	40.29	31.11	39.40	43.97	47.02
(2n+1)-grams	47.89	54.70	43.21	53.68	51.32	53.46

The rate of precision is increased for the most frequent sense, it is explained by the fact that the number of samples containing the most frequent sense is more important that the number of samples containing the other senses.

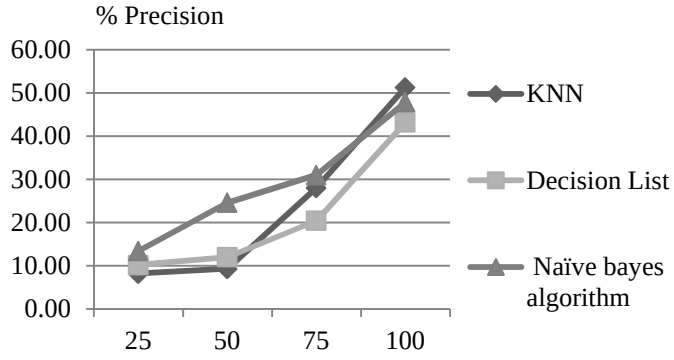

Fig. 1. Obtained results by the different algorithm depending on the amount of data considered in the training phase.

Since that during the step where we have to count the number of occurrences of each collocation contained in the original sentence, we take into consideration all the derivation of the words using the khoja stemmer. For example for the word "قرأ" "karaa" that occurs with the word "كتب" "kataba", we have to count the number of occurrences of "قرأت, يقرؤون, قرأت, قرأتنّ, تقرأ يقرأ, …" in the extracted samples.

We detail in Table 4, the rate of precision obtained with and without the stemming. The most supervised methods estimate the probability of each word using the context of the previous n-1 words. The problem of those methods is that unfortunately it assigns zero to n-grams that have not been observed, in the training phase. To avoid this problem we have to smooth the zero counts (see Section 4.1).

In Table 4, we give also the results of the different methods without and with the use of the smoothing for the (2n+1) grams experiment.

From the results cited in table 4, we find that the stemming increase the precision by a percentage that varies between 9% and 21% for the different methods. It was supposed that the smoothing will decrease the rate of precision, because it increases all probabilities for unseen words.

TABLE IV
RATE OF PRECISION OBTAINED BY CONSIDERING THE STEMMING AND THE SMOOTHING STEPS.

Tests / Methods	Without stemming and smoothing	Without Stemming and with smoothing	With stemming and without smoothing	With smoothing and stemming
Naïve Bayes	31.25	31.75	47.89	48.23
Decision List	22.06	22.56	43.21	43.86
KNN	42.19	42.69	51.32	52.02

Also the smoothing increases the precision and recall of about 0.5%, and this increase is encouraging to perform the disambiguation quality.

As we have cited in the beginning of this section, the supervised methods needs a highly amount of data. The results varies from a word to another one and for the majority of the tested words the k nearest neighbor algorithm gives the better results and for some other words ("عين،كتب، شعر" and their corresponding transcription is "ayn, kataba, chaar") the Naïve Bayes algorithm is the best one. The rate of precision obtained by the decision list is in all the case is more less than the rate obtained by the Naïve Bayes algorithm. We may explain these results by the fact that the other supervised algorithm needs a so large training sets than the Naïve Bayes algorithm.

V. COMPARISON WITH SOME RELATED WORKS

We can cite the work of Mona Diab that uses a supervised learning approach called "bootstrap"[15]. This approach is highly accurate in the average of 90% of the evaluated data items based on Arabic native judgment ratings and annotations. Also, we find the work of Elmougy[16], where the Naïve Bayes algorithm was applied for the Arabic language. Some pretreatement steps were applied like word rooting and eliminating stopwords, since that they use the net and a dictionary to collect ten training samples to each word for the testing phase. This work achieves a rate of precision of 73%. Compared to our work the amount of data is more less, and collecting the testing samples from the net is a hard task and not sufficient.

Finally, Soha M. Eid [17] compared the Rocchio Classifier to Naïve Bayesian classifier, the most frequent sense and the support vector machine using arabic lexical samples. The Rocchio classifier achives an overall accuracy of 88% as the best rate and reduce the error by over 14%. But they test only five ambiguous words and they haven't explain how tagging the samples of the training phase.

Compared to our work we obtain a more less rate of precision because of the important number of ambiguous tested words (fifteen ambiguous words). Also as a comparative study there is no test for the influence of the window size, the stemming and the smoothing on the quality of disambiguation.

For the other English related works, we can cite the experimental study that compares some supervised algorithms to disambiguate six senses of the word line [18] and [19]. Also the work of Pedersen where he compared the Naïve Bayes with Decision tree, Rule based learner, etc, to disambiguate the word line and 12 other words [20]. All these works, found that the Naïve Bayes algorithm performed as well as the other supervised algorithms, which is the same results founded in this work. Compared to the number of tested words by the English related works, we have to point that we test fifteen words for a derivational language that suffers from the lack of resources.

We can also compare the obtained results by some works of unsupervised Arabic word disambiguation, where the same samples and the same words were tested. In the first work [21], it was proposed to use some information retrieval measures with the Lesk algorithm and it achieves a rate of 73%. In the second one [22], a Context matching algorithm returns a semantic coherence score corresponding to the context of use that is semantically closest to the original sentence. This algorithm achieves a precision of 78%. In this work, we obtain a less rate of precision. We can presume that the supervised works are more satisfactory for the task of Arabic Word Sense Disambiguation.

VI. CONCLUSION

This paper has presented an experimental study of some supervised algorithms that were applied to perform word sense disambiguation in Arabic. These algorithms are based on tagged samples and a very important amount of data in the used corpus.

For a sample of fifty ambiguous Arabic words that are chosen by their number of senses out of contexts, the KNN achieves the best performance. We conclude that the supervised methods need an important amount of tagged data to achieve satisfactory results. We propose in future works to integrate some other resources and experiment some other supervised methods.

REFERENCES

[1] R. Mihalcea, "Word Sense Disambiguation Using Pattern Learning and Automatic Feature Selection", in Journal of Natural Language and Engineering (JNLE), December 2002, p.p: 348–358.

[2] H. T. Ng and H. B. Lee, "Integrating multiple knowledge sources to disambiguate word senses: An examplar-based approach". In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA, 1996, p.p: 40–47.

[3] L. Al-Sulaiti, E. Atwell, "The design of a corpus of contemporary Arabic". International Journal of Corpus Linguistics, vol. 11, 2006, pp. 135-171.

[4] M. Ben Mukarram and al-Ifriqi al-Misri ibn MANZUR, " Lisàn al-'arab", Ibn Manzûr, 15 volumes, 1956, Beyrout.

[5] J. Savoy, Y. Rasolofo, "Report on the TREC-11 Experiment: Arabic, Named Page and Topic Distillation Searches". Eleventh Text Retrival Conference TREC, 2002.

[6] C. Fox, "A stop list for general text". SIGIR Forum, 1990, Vol. 24, No. 1-2, pp. 19-35.

[7] A. Chen, F. Gey, translation Term Weighting and Combining Translation Resources in Cross-Language retrieval, Tenth text retrieval conference, 2001, TREC.

[8] S. Gerard, M.J. McGill, “Introduction to modern information retrieval”, ISBN: 0070544840, 1983, p.p: 448.

[9] K. Shereen and G. Roland, “Stemming Arabic text”, Computer Science Department, Lancaster University, Lancaster, UK, 1999.

[10] R. Navigili, “Word Sense Disambiguation: A Survey”. ACM Computing Surveys, Vol. 41, No. 2, Article 10, Publication date: February 2009.

[11] T. Pedersen, “Learning probabilistic models of word sense disambiguation”, Ph.D. dissertation. Southern Methodist University, Dallas, TX. 1998.

[12] D. Yarowsky, “Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French”. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (Las Cruces, NM), 1994, p.p: 88–95.

[13] A. Zouaghi, L. Merhbene, M. Zrigui, “Word Sense disambiguation for Arabic language using the variants of the Lesk algorithm”, WORLDCOMP’11, Las Vegas, juillet 2011, p.p. 561-567.

[14] D. Yarowsky, “One sense per collocation”. In Proceedings of the ARPA Workshop on Human Language Technology, Princeton,1993, pp. 266-7.

[15] M. Diab and P. Resnik, “An unsupervised method for word sense tagging using parallel corpora”. Proceedings of the ACL40th Meeting of the Association for Computational Linguistics, Philadelphia, U.S.A. 2002, pp. 255-262.

[16] S. Elmougy, H. Taher and H. Noaman “Naïve Bayes Classifier for Arabic Word Sense Disambiguation”. In proceeding of the 6th International Conference on Informatics and Systems, 2008, pp: 16-21.

[17] M. Soha Eid, et al., “Comparative Study of Rocchio Classifier Applied to supervised WSD Using Arabic Lexical Samples”. Proceedings of the tenth conference of language engeneering (SEOLEC’2010), Cairo, Egypt, December 15-16, 2010.

[18] C. Leacock, G. Towell and E. Voorhees, “Corpus based statistical sense resolution”. In Proceedings of the ARPA Workshop on Human Language Technology, 1993, p.p. 260-265.

[19] R.J. Mooney, “Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. Proceedings of EMNLP, 1996, p.p: 82-91.

[20] T. Pedersen, “Learning Probabilistic Models of Word Sense Disambiguation”. Ph.D. Dissertation. Southern Methodist University, 1998.

[21] A. Zouaghi, L. Merhbene and M. Zrigui, “Combination of information retrieval methods with LESK algorithm for Arabic word sense disambiguation”. Journal Article published in the Artificial Intelligence , Online First, 30 May 2011, Review; DOI: 10.1007/s10462-011-9249-3.

[22] L. Merhbene, A. Zouaghi and M. Zrigui, Ambiguous Arabic Words Disambiguation. In Proceeding of The 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD’10), The University of Greenwich, London, United Kingdom, 9-11 June, 2010, p.p. 157-164.