A Semi-Supervised Method for Arabic Word Sense Disambiguation Using a Weighted Directed Graph

Laroussi Merhbene
LATICE / Faculty of sciences of Monastir, Monastir, Tunisia
Aroussi_merhben@hotmail.com

Anis Zouaghi
LATICE, ISSAT Sousse, University of Sousse, Tunisia
Anis.zouaghi@gmail.com

Mounir Zrigui
LATICE, Faculty of sciences of Monastir, Monastir, Tunisia
mounir.zrigui @fsm.rnu.tn

Abstract

In this paper, we propose a new semi-supervised approach for Arabic word sense disambiguation. Using the corpus and Arabic Wordnet[1], we define a method to cluster the sentences containing ambiguous words. For each sense, we generate a cluster that we use to construct a semantic tree. Furthermore, we construct a weighted directed graph by matching the tree of the original sentence with semantic trees of each sense candidate. To find the correct sense, we use a similarity score based on three collocation measures that will be classified using a novel voting procedure. The proposed method gives a high rate of recall and precision.

1 Introduction

The human language is so complex to be learned. The syntactic form of words, the relation between a specific word form and its meaning are the basic parts of an intelligent system for the natural language processing.

In this work we aim to solve the task of identifying the sense of the ambiguous word. This task is called Word Sense Disambiguation (WSD), which is one of the oldest problems in natural language processing (NLP) (Agirre and Edmond, 2006). This work is part of a general frame-work of Arabic speech (Zouaghi, 2008).

In this work, we combine a supervised and an unsupervised method for Arabic word sense disambiguation. The innovative part in this work is the construction of a semantic tree for each sense of the ambiguous word. Also we define a voting procedure that gives a weight for the score measure.

This paper is organized as follows. Section two describes the proposed method. The experimental results are described in section three. Finally this paper is concluded in section four.

2 Proposed method

We propose a semi-supervised method for Arabic word sense disambiguation.

For the unsupervised part of the proposed method, we use Arabic Wordnet (Black et al., 2006) and the corpus to construct sense clusters (group of sentences) characterizing a specific sense of the ambiguous word. Furthermore we construct a semantic tree for each sense of the ambiguous word.

The disambiguation procedure is based on the step of matching the semantic tree with the tree of the original sentence. We use a score measure (based on three collocation measures) to find the closest semantic tree to the tree of the sentence to be disambiguated.

The supervised part uses a voting procedure that will rank collocation measures during a classification task. The sense given by the measure having the highest rank will be attributed to the ambiguous word. In what follows we describe with more details each step cited above.

2.1 Construction of the sense clusters

In the first we apply some pre-treatment steps to glosses of the ambiguous word (definitions and synonyms extracted from Arabic wordnet) and sentences containing the ambiguous word (collected from the used corpus). Using the Kho-

[1] Arabic Wordnet is a concept dictionary with mappings between word definitions.

International Joint Conference on Natural Language Processing, pages 1027–1031,
Nagoya, Japan, 14-18 October 2013.

ja stemmer and the approximate string matching, we are able to construct the sense clusters. Some pre-treatment steps will be applied to these clusters. In what follows, we detail the steps of the proposed method.

Pre-treatment: Using the corpus we collect sentences containing the ambiguous word, we have to search the root of the ambiguous word (exp: for the word "العين" "alayn" we have to search the root "عين" "ayn"). The segmentation of these sentences is based on punctuation (., ;, !, ?, etc.) and on the number of the words that have to be more than three.

Subsequently, we eliminate the stop-words that occur frequently in the corpus and they have no significant relation to the sense of the word. We use a general stop-list containing 29,985 stop-words, this list were elaborated by Arabic linguistics and judged as sufficient for the task of WSD.

Root extraction: We use the Khoja stemmer (Khoja,1999) for words contained in the glosses of the ambiguous word. Its advantage is that it uses a large linguistic data such as the list of verbal and noun patterns, stop-words, list of diacritic characters, etc.

For a specific word, this stemmer extracts the longest suffix and prefix, which will be matched with the existing list of patterns to extract the root. We notice that we use the list of stop-words in addition to the already used list (detailed in the previous paragraph).

Sense Clustering: The basic idea of Sense clustering is that the sentences representing the meaning of a particular sense are grouped in the same cluster C_i (Cluster of the i^{th} sense of the ambiguous word).

The list of sentences extracted from the corpus will be classified into clusters using the roots of the words containing in each gloss. To find the possible occurrences of the roots, we use the approximate string matching algorithm (Elloumi, 1998).

In the first we fill a matrix of the two words to be compared w_i and w_j. After that we use the step of back-tracking, to find the shortest common subsequence.

The words containing the common subsequence will be considered as occurrences of the stem. The Sentences containing the occurrences of stems obtained from glosses are grouped into clusters representing each sense of the ambiguous word.

2.2 Semantic Tree construction

A text can be represented by Trees or graphs (co-occurrence graphs (Agirre and Sorora, 2007), collocation graphs (Klapaftis and Manandhar, 2008), semantic graphs (Plaza and Diaz, 2011)) that differs in the structure of text representation.

The first step is to transform the sentences of the clusters to binary trees, T = (N, E, R, RC, LC, L), where:

- N is a set of nodes, N = {n_1… n_2}. Each node corresponds to a concept in the binary Tree.
- E is a set of edges that represents the relation between the node n_i to the node n_j.
- R is the root of the tree which is the ambiguous word. RC is the set of right children which are the words occurring on the right of the ambiguous word.
- LC is the set of left children which are the words occurring on the left of the ambiguous word.
- L is a function assigning the level of the nodes, it corresponds to their position regarding the ambiguous word.

Expect the root, each node of the tree has exactly one child. We denote <R, RC, LC> a binary tree.

The second step is to merge all the obtained trees corresponding to the sentences contained in the same cluster. Accordingly, we obtain a semantic tree, ST = (N, E, R, C, L, Nb, H), where:

- C is the set of merged nodes, C={c_1,…c_n}. The right and left child of each binary tree will be linked to the root of the semantic tree.
- Nb is a function that returns the number of nodes in the semantic tree.
- H is a function that returns the height of the semantic tree.

The step of merging trees uses an algorithm of breadth-first traversal, to find the repeated node that may have a higher level, same level or a lower level.

2.3 WSD procedure

In the first, we apply some pre-treatment steps to the original sentence containing the ambiguous word. The process of disambiguation is based on three steps:

Step 1: Weighted directed graph construction: We add edges weighted by the collocation measures between the nodes N_i of the tree of the original sentence (called T_{os}) and the nodes N_j of the semantic tree of each sense (called $\mathbf{ST}_{S_k}$, where S_k corresponds to the k^{th} sense).

This step called matching allows us to obtain a weighted directed graph. After eliminating stopwords, we extract the roots of the words contained in the original sentence. These roots are the nodes of the tree and the level in the tree $T_{os}(N)$ will be affiliated corresponding to their position regarding the ambiguous word.

Each node of the tree extracted from the original sentence is matched with the nodes of the same level in the semantic tree of a particular sense. The links used for the matching step appear as a dashed line. They are weighted using one of the three collocation measures (Maning and Schütze, 1999) detailed in what follows:

The T-test

The T-test is measured as follows (see equation 1).

$$wc_{ij} = T = (\bar{x} - \mu) / (\sqrt{\frac{s^2}{N}}) \qquad (1)$$

The mean of the distribution μ is measured by multiplying $P(w_i)$ to $P(w_j)$, where $P(w)$ = number of occurrences of w in the corpus / Total number of words in the corpus. $\bar{x}$ (sample mean) is equal to s^2 (sample variance), measured by dividing the number of occurrences of the two words together by the total number of words in the corpus.

The Mutual Information

This measure determines how much a word can be informative for another word. The mutual information is measured as follows (see equation 2):

$$wc_{ij} = MI = \log_2 \frac{P(wi,wj)}{P(wi)\,P(wj)} \qquad (2)$$

The Chi-Square $\chi 2$

The equation 3 in what follows details the measure of $\chi 2$.

$$wc_{ij} = \chi\, 2 = \frac{N \times (C_{1,1} \times C_{2,2} - C_{1,2} \times C_{2,1})^2}{(C_{1,1} + C_{1,2}) \times (C_{1,1} + C_{2,1}) \times (C_{1,2} + C_{2,2}) \times (C_{2,1} + C_{2,2})} \qquad (3)$$

The basic principle is to count $C_{1,1}$ (the number of occurrences of w_i and w_j together), $C_{1,2}$ (the number of occurrences of w_i without w_j), $C_{2,1}$ (the number of occurrence of w_j without w_i) and $C_{2,2}$ (the number of bigrams in the corpus that don't contains w_i or w_j).

Step 2: Semantic similarity measure: We define a score measure that allows us to choose the closest semantic tree ST_{S_k} to the tree of the original sentence T_{OS}. The score measure is defined in what follows (see equation 4).

$$\text{Score} = \textstyle\sum_{N_i \in T_{os}} (\sum_{\mathrm{N}_j \in \mathrm{ST}_{S_k}} (\frac{\mathrm{wc_{ij}}}{\mathrm{ST}_{S_k}(\mathrm{L}(\mathrm{N_j}))}) / \mathrm{Nb}(\mathrm{ST}_{S_k})) / \mathrm{Nb}(\mathrm{T_{OS}})) (4)$$

The score measure is the average of the product between nodes of ST_{S_k} and T_{os}. Where $\mathrm{Nb}(\mathrm{T_{OS}})$ is the total number of nodes in T_{os} and $\mathrm{Nb}(\mathrm{ST}_{S_k})$ is the total number of the nodes linked to each node of ST_{S_k}. $\mathrm{ST}_{S_k}(\mathrm{L}(\mathrm{N_j}))$ corresponds to the level of the node N_j contained in the semantic tree ST_{S_k}.

As a result we give the sense that corresponds to the semantic tree that obtains the highest score.

The weights obtained by the collocation measures wc_{ij} are normalized to low weights between 0 and 1.

Step 3: Voting procedure: The idea is that during the classification task, we ranked measures of collocation according to the correct attribution of the sense.

In the case where the three collocation measures agree on the same result, then the given sense will be attributed to the ambiguous word and the rank of the collocation measure will not be changed.

In the case where more than one measure agrees on the attribution of a sense, then, we have to choose the sense having the majority of votes. The rank of the measures that vote for the attributed sense will be increased and the rank of the other measures will be decreased.

The final case is where all the measures do not give the same result. The result given by the measure having the highest rank (attributed during the last N tests) will be used to attribute the sense of the ambiguous word. In what follows, we detail results given by the described method.

3 Experimental Results

3.1 Used resources and tested data

Due to our need to maximize the keywords that define a specific sense, we use Arabic Wordnet (AWN) (Black et al., 2006) which is a dictionary. Words are arranged semantically instead of alphabetically. Synonymous words are grouped together to form synonym sets.

Also we collect a large corpus from newspaper articles, which were recorded from different corpus that are available on the net. In total, we collect a corpus that counts 123,854,642 words.

For the missed senses in the corpus, we collect from the net the contexts containing these senses and we added them to the used corpus.

3.2 Obtained Results

In the table 1 below, we report the statistics of the tested data and the obtained rate (Precision, Recall, F-Score) given by the voting procedure (VP) and the collocation measures for 127 ambiguous words. In total we test 42,316 sentences.

For each sense we test 40 sentences. For the classification part of the voting procedure, we use 20 samples per sense (labeled data). In total, we have 4,582 tagged samples. We haven't found an important difference between the sense tags, the agreement between the annotators is in the average of 95%.

wc_{ij}	#correct disambiguated sentences	Recall	Precision	F-Score
T_{test}	31,298	0,739	0,754	0,747
MI	29,783	0,703	0,718	0,710
χ 2	32,122	0,759	0,774	0,766
V P	35,145	0,830	0,830	0,830

Table1. Performances of our method.

We remark that the F-score obtained by applying the voting procedure is higher than those obtained by any one of the collocation measures.

There is not a big difference between the Precision and the Recall obtained by any of the used collocation measures. This can be explained by the fact that the majority of the tested words were disambiguated. However, the best collocation measure is the χ 2, otherwise the voting procedure increases the F-score by 6,4%.

We measure the performance of our method under the number of nodes in ST. The obtained results indicate that for semantic trees with at least 500 nodes, the performance of our method increases consistently. However, the F-Score reaches the top and becomes stable for semantic tree sizes between 2,000 and 3,000 nodes. We conjecture that more the semantic tree is enriched by the nodes, more the F-Score increases.

3.3 Comparison with other works

In order to contextualize the obtained results in the current state of the art, fifty ambiguous words that are used in the experimental study of this work were evaluated in previous works of Arabic WSD:

- Supervised works which are the naïve bayesian algorithm, the Decision List and the K Nearest Neighbor (Merhbene et al., 2012).
- Based knowledge works which are the original Lesk algorithm and the modified Lesk algorithm that uses Arabic Wordnet and five similarity measures (Zouaghi et al., 2011).
- Unsupervised work for Arabic WSD based on a combination between some information retrieval measures and the Lesk algorithm (Zouaghi et al., 2012) and (Merhbene et al., 2010).

Compared to our method, we note that the Lesk algorithm is limited to dictionary definitions that we use. Therefore, the absence of certain words can radically change the results. The modified Lesk algorithm using the Leacock and Chodorow measure (Leacock and Chodorow, 1998) is the most performed between based knowledge methods with a rate of Precision equal to 67,73%.

The supervised methods need an important amount of tagged data to achieve satisfactory results. They need to be applied in specific domains. The K nearest neighbor algorithm achieves the best rate of Precision (52,02%).

Finally, compared to the unsupervised method of Arabic WSD, the rate of precision is enhanced by 10% using more 117 ambiguous words.

4 Conclusion and future work

This paper describes a novel approach for the disambiguation of the Arabic language based on the weighted directed graph.

During the step of disambiguation, we match the tree of the sentence to be disambiguated with each semantic tree of the senses candidate. The obtained weighted directed graph uses three collocation measures that will be classified using a novel supervised voting procedure. Results show that our method achieves a very high recall and precision (83%).

In the future works, we propose to test more ambiguous words, using more tested data and resources to confirm the positive obtained results.

References

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

Agirre E. and Sorora A. 2007. *A graph based unsupervised system for induction and classi-fication.*

The Fourth International Workshop on Semantic Evaluations, p.p: 346-349.

Black, W., Elkateb, S., Rodriguez, H., Alkhalifa, M., Vossen, P., Pease, A. and Fellbaum, C. 2006. *Introducing the Arabic WordNet Project*, in Proceedings of the Third International WordNet Conference, Sojka, Choi, Fellbaum and Vossen eds.

Elloumi M. 1998. *Comparison of Strings Belonging to the Same Family*. Information Sciences, An International Journal, Elsevier Publishing Co., Amsterdam, North-Holland (Publisher), 111(1-4), p.p:49-63.

Klapaftis I, and Manandhar S. 2008. *Word Sense Induction Using Graphs of Collocations*. In the proceeding of the 18th European Conference On Artificial Intelligence, p.p: 298-302.

Khoja, Shereen, 1999. Stemming Arabic Text. http://zeus.cs.pacificu.edu/shereen/research.htm

Leacock C. and Chodorow, M. 1998. *Combining local context and WordNet sense similarity for word sense identification*. MIT Press, Cambridge, Massachusetts, p.p: 265-283.

Manning C. and Schütze H. 1999. *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge.

Merhbene L., Zouaghi A. and Zrigui M. 2012. *Arabic Word Sense Disambiguation*. In Proceeding of International Conference on Agents and Artificial Intelligence, Volume 1, Valencia, Spain, 22-24 January, p.p:652-655

Merhbene L., Zouaghi A. and Zrigui M. 2012. *Lexical Disambiguation of Arabic Language: An Experimental Study*. The Journal Polibits Vol 46, pp: 49-54.

Plaza L. and Diaz A. 2011. *Using Semantic Graphs and Word Sense Disambiguation Techniques to Improve Text Summarization*. The Procesamiento del Lenguaje Natural, p.p: 97-105.

Zouaghi A., Merhbene L., Zrigui M. 2012.*Combination of information retrieval methods with LESK algorithm for Arabic word sense disambiguation*. Journal Article published in the Artificial Intelligence Review. Volume 38, Issue 4, DOI: 10.1007/s10462-011-9249-3; Online ISSN: 1573-7462, p.p:257-269.

Zouaghi A., Zrigui M. and Antoniadis G. 2008. *Understanding of the Arabic spontaneous speech: A numeric modelisation*, Revue TAL VARIA.

Zouaghi A., Merhbene L., Zrigui M. 2011. *Word Sense disambiguation for Arabic language using the variants of the Lesk algorithm*, in Proceeding of the International Conference on Artificial Intelligence (ICAI'11), Las Vegas, USA, pp: 561-567.