Intelligent Information Management, 2009, 1, 122-127
doi:10.4236/iim.2009.12018 Published Online November 2009 (http://www.scirp.org/journal/iim)

Scientific Research

Word Sense Disambiguation in Information Retrieval

Francis de la C. Fernández REYES, Exiquio C. Pérez LEYVA, Rogelio Lau FERNÁNDEZ
Instituto Superior Politécnico, José Antonio Echeverría, Marianao, Cuba
Email:{*ffernandez, exiquio, lau*}*@ceis.cujae.edu.cu*

Abstract: The natural language processing has a set of phases that evolves from lexical text analysis to the pragmatic one in which the author's intentions are shown. The ambiguity problem appears in all of these tasks. Previous works tries to do word sense disambiguation, the process of assign a sense to a word inside a specific context, creating algorithms under a supervised or unsupervised approach, which means that those algorithms use or not an external lexical resource. This paper presents an approximated approach that combines not supervised algorithms by the use of a classifiers set, the result will be a learning algorithm based on unsupervised methods for word sense disambiguation process. It begins with an introduction to word sense disambiguation concepts and then analyzes some unsupervised algorithms in order to extract the best of them, and combines them under a supervised approach making use of some classifiers.

Keywords: disambiguation algorithms, natural language processing, word sense disambiguation

1. Introduction

The natural language processing involves a set of tasks and phases that evolves from the lexical text analysis to the pragmatic one in which the author's intentions are shown. One natural language problem is ambiguity, as we can see in the following sentence: "I made her duck". This is a classical example of ambiguity; someone who hears this phrase understands the speaker's intention, but it is harder make the computer understands it. First, the words duck and her are morphologically or syntactically ambiguous in their part-of-speech. Duck can be a verb or a noun, while her can be an object pronoun or a possessive adjective. Second, the word make is semantically ambiguous; it can mean create or cook. Finally, the verb make is syntactically ambiguous in a different way. Make can be transitive, that is, taking a single direct object, or it can be intransitive, that is, taking two objects, meaning that the first object (her) got made into the second object (duck). Finally, make can take a direct object and a verb, meaning that the object (her) got caused to perform the verbal action (duck) [1].

There are already a lot of works that resolve, almost complete, the lexical ambiguity, as part of syntactic analysis in natural language processing. However, a current problem, without a complete solution yet, is semantically ambiguity, according to Alexander Gelbukh and Grigori Sidorov [2]. Nowadays, algorithms that attempt resolving this problem are divided into two groups: discriminative algorithms and disambiguation algorithms.

One of the natural language processing applications is the information retrieval. On the one hand, Internet and digital libraries have a huge amount of knowledge that can answer a lot of questions that people may have. On the other hand, the amount of information is so huge that interferes in its proficiency because it is impossible to process it easily. At present, more used techniques for information retrieval implicate the search of keywords: Files that contain the words that the user indicates are being found. Another idea to set relevant knowledge, in front of a question, will be attending to synonym relations to establish similar documents; besides, the use of relations between words under a specific context create the necessity of employ a word sense disambiguation algorithm to understand the sense of the words that user is using in his search.

This paper presents an analysis of the existing disambiguation algorithms and it analyses the quality of each of them taking into account the metrics that have been establish for the evaluation. At the same time, it shows a possible combination of features and classifiers to propose a word sense disambiguation algorithm that resolves some deficiencies detected before and improve the evaluation parameters.

2. Word Sense Disambiguation Algorithms

The word-sense disambiguation process consists of assigning to each given word in a context, one definition or meaning (predefine sense or not), that is distinguishable from others that it can have.

The disambiguation techniques may be classified into

the way that is shown in Figure 1.

In the case of discrimination algorithms, a meaning for the word is not enough, it is necessary to determine which occurrences have the same sense, without the need of establish which it is. Besides, they work with no linguistic resources. On the contrary, disambiguation algorithms reach the meaning of the word using external linguistic resources (corpus or knowledge bases). The discrimination algorithms identify context vectors for all given word occurrences, divide the vectors into groups and interpret each of them as a sense.

Corpus based methods (supervised) collect a set of examples, manually tagged, for each sense of disambiguation words and induce a classifier (Support Vector Machine, Näive Bayes [3] from these examples, then the disambiguation is reduced to the process of classifying the word in one of his possible senses. Among the limitations you can find in those methods we have the knowledge domain dependency (due to the example set use) and the manual tagging is extremely expensive.

Knowledge based methods (not supervised) do not require tagged corpus, rather use external linguistic resources, and therefore, the disambiguation can be considered at any knowledge domain, as long as that external resource accept it. The supervised methods obtain better results than the not supervised ones, but they prefer the seconds because they are not restricted to a specific knowledge domain. Knowledge based methods can make use of dictionaries as Lesk's algorithm [4], of thesaurus as Yarowsky's [5] and of WordNet as Resnik's [6].

As not supervised algorithms can be applied to any knowledge domain, as long as that external resource allows it, the authors analyze just them in the rest of the document. Specifically we will examine the following methods: based on grouping, Fuzzy Borda Voting, Extended Lesk, Conceptual density and Sense probability. There are others algorithms that also result interesting, but we don't analyze them in the present paper, examples of these are: Meaning affinity model [7], using automatically acquired predominant senses [8] and based on lexical cohesion [9].

2.1. Using Sense Clustering for Word Sense Disambiguation

This method doesn't require the use of a training set, just use WordNet. The algorithm begins to grouping all senses of the disambiguation target words. This process tries to identify cohesive senses' groups; those are created by means of Extended Star cluster algorithm and β_0-similarity graph between different senses [10]. Then, the method filters the groups to select those that match the best with the context. If the selected groups disambiguate all the words (each group show only one sense), then the process stops and the senses belonging to the selected groups are interpreted as the disambiguated ones. Otherwise, the clustering and filtering steps are performed again (regarding the remaining senses) until the disambiguation is achieved or when it is impossible to raise β_0 threshold [10].

The algorithm input is the disambiguation target words set W and the context represented as topic signature T [10].

There are two sub process cluster and filter. The first one is carried out by the Extended Star Clustering Algorithm, which builds star-shaped and overlapped clusters. Each cluster consists of a star and its satellites, where the star is the sense with the highest connectivity of the cluster, and the satellites are those senses connected with the star. The connectivity is defined in terms of the β_0-similarity graph, which is obtained using the cosine similarity measure between topic signatures and the minimum similarity threshold β_0.

Once clustering is performed over the senses of words in W, a set of sense clusters is obtained. As some clusters can be more appropriate to describe the semantics of W than others, they are ranked according to a textual measure context T.

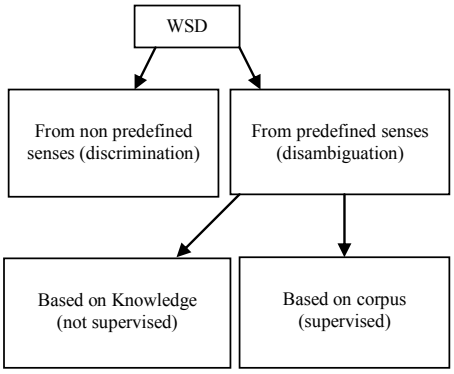

Figure 1. Classification of word sense disambiguation algorithms

After grouping the algorithm is obtained a set S with the senses of the selected group. Then, the method compares if the sense quantity in S matches the word quantity in W, if this is true, the disambiguation process stops, on the contrary, the cluster algorithm starts again with other threshold β_0.

The algorithm behaves adequately in front of nouns instances, although it shows an inconvenience, it does not disambiguate proper nouns due to the lack of senses in WordNet. Although, it behaves in a non correctly way for verbs, of 304 tested, the algorithm just disambiguated the 30% in a correct way. This result is a consequence of the high grade of polysemy that verbs had and the few relations that appear between them in WordNet, therefore, the cluster with high polysemy level can not create cohesive groups and fail the disambiguation process. Besides, the context is reduced to the sentence, conesquently, become interesting extending the context to paragraph, this approach includes more verbs and the search of WordNet relations increases, the effect is the improvement of the clustering process.

2.2. Combining Different Methods by Means of Fuzzy Borda Voting

The original scheme, Borda voting score, was introduced in 1770 by the mathematician Jean Charles de Borda of the French Academy. This method consists of a pondering system, that fight the general believes that the candidate which obtains the majority of votes is the one that voters prefer and show examples of contradictions in votes' system used before.

Borda voting score establishes a punctuation system where each voter gives a mark to each one of the candidates, following the order of their preferences. In order to find the winner, add the obtained scores for each candidate. This system has an inconvenience it does not perform rules for the weights of the candidates, which could imply an arbitrary assignment that changes the result.

The fuzzy variant allows the experts (voters on the original scheme) gives a numerical value that indicates how some alternatives (candidates on the original scheme) are preferred among others, evaluating the preferences in a range between 0 and 1. In Rosso & Buscaldi [11], each expert gives a mark to each alternative, according to the number of alternatives worse than it. The algorithm establishes the use of experts (other disambiguation methods) to achieve the disambiguation process, and the ones considered are: Sense probability, Extended Lesk, Conceptual density (for verbs) and WordNet domains.

To obtain fuzzy preferences relations, the output weights $w_1, w_2, ..., w_n$ of each expert k are transformed to fuzzy confidence values by means of the following transformation [11]:

$$r_{ij}^k = \frac{w_i}{w_i + w_j}$$

where r^k_{ij} is considered as the degree of confidence with which the expert k prefers alternative x_i to x_j.

With the fuzzy preferences relations of m experts over n alternatives x1, x2, . . ., xn. For each expert k we obtain a matrix of preference intensities [11]:

$$\begin{pmatrix} r_{11}^k & r_{12}^k & \cdots & r_{1n}^k \\ r_{21}^k & r_{22}^k & \cdots & r_{21}^k \\ \cdots & \cdots & \cdots & \cdots \\ r_{n1}^k & r_{n2}^k & \cdots & r_{nn}^k \end{pmatrix}$$

The final value assigned by the expert k to each alternative xi coincides with the sum of the entries greater than 0.5 in the i-th row in the preference matrix.

Therefore, the definitive fuzzy Borda count for an alternative xi is obtained as the sum of the values assigned by each expert *k* [11]:

$$r(x_i) = \sum_{k=1}^{m} rk(x_i)$$

In order to apply this system on word sense disambiguation the first thing to do is determine the target word and the senses set of it, afterward, using an expert (for example, "Sense probability"), calculate the weights of each sense and begin the competition between them. Establish the confidence degree r^k_{ij} of the first sense with the second one, then with the third one, and so on in order to make the competition. These fuzzy values are part of the first row of the preferences matrix, each row of the matrix match an analyzed sense. In this way are go establishing the different confidence degrees with the other senses till complete the matrix. To end with the expert, establish the value assigned to the sense considered, this value is calculated adding the row that matches with the alternatives proposed, but avoiding the value founded in matrix principal diagonal because it is the alternative analyzed with it self. Then proceed in same way with the other experts, at the end add, for each sense, the values that each one of them assigned and the biggest score is the winner, considering this one as the correct sense of the target word.

The algorithm has as principal idea the usage of a voting system with fuzzy bases where each expert is word sense disambiguation algorithm knowledge based, these is a positive issue, the approach put under competition various algorithms and the correct sense is that one who receive the biggest score. The ambiguity resolution for verbs behaves adequately, the experts that are considered work in the sentence context. Will be interesting find ano-

ther set of experts, this selection should be doing according to the algorithm results taking into account the part of speech and considering also a baseline as expert, then realize tests changing the competition scheme to analyze the behavior of disambiguation process.

2.3. Sense Probability (Most Frequently Sense)

In nineties grow up an interest on the line of the establishment of methods for automatic word sense disambiguation evaluation that offers a reference point to measure the quality of done work. A starter point is the set of a superior and inferior limit for the disambiguation results: the inferior limit match the choice of the most frequently sense, while the superior limit match the human tagged. The authors asseverate as correct choice the most frequently sense in a 75% of the cases. About superior limit, human tagged, exist many controversies about his reliability, fed by contradictory results obtained in several experiments. Other contribution is the morphsyntactic and semantic tagged words of open class from Brown corpus, making benchmarks for disambiguation systems in senses choosing: the chance, the frequency (most frequency sense), the concurrence [12].

Those basic disambiguation techniques, that usually not imply any kind of linguistic knowledge, are used as a reference point for the evaluation of those systems [12].

One of the baseline measure used is Sense probability. It consists in assign the first sense of WordNet to each target word and it is calculated as follow:

$$BL_{MFS} = \frac{1}{|T|}\sum_{i=1}^{|T|}\delta(w_i,k)$$

where $\delta(w_i,k)$ is equal to 1 when the k-th sense of the word w_i belongs to the group manually tagged by the lexicographer (example, SemCor) for word w_i and it is 0 otherwise. The Most Frequency Sense (MFS) calculations are based on the frequencies of SemCor corpus terms. T is the test corpus where are contained the instances w_i. WordNet rank the senses taking into account the appearing frequencies of them inside SemCor corpus. This baseline always resolves the semantic ambiguity problem with a 78% precision [12].

This technique lack totally of linguistic information, however, is a good idea has a tagged corpus to find the most frequently sense. Actually, it is not a knowledge based disambiguation algorithm but it is used as a reference point to compare the algorithm done.

2.4. Extended Lesk Algorithm

This algorithm is an improvement version (WordNet based) of the well known Lesk procedure and based it on the use of dictionaries. The original algorithm was supported in the comparison of the gloss target word with the context words and their gloss. The improvement consists in taking into account also the gloss of the concepts related to the target word, by means of various WordNet relations [13].

Then, the similarity between a word sense and his context is calculated by overlapping. Overlapping is the intersection between to synsets set (a synset is a number that implies sense for WordNet), on one hand, the ones for the target word and on the other hand the gloss of the context synsets. To the target word is assigned the sense obtained that betters overlap with the gloss of the context words and their related synsets [13].

The heterogeneous approach of the Extended Lesk algorithm show better results than the homogeneous ones, consider the part of speech not improve the precision, excluding the case of adjectives. The context of this algorithm is a window of size 7, 9 or 11 around the target word, does not take into account the sentence context and it uses various WordNet relations and glosses. The size of the window depends directly from the part of the speech, while biggest is the window more decrease the polysemy and reach better precision results, most of all in verbs.

2.5. Conceptual Density Algorithm

Conceptual density was originally introduced by Agirre and Rigau in 1996 above WordNet. It is calculated over sub-graphs of this lexical database, determined by hypernymy relations. The proposed formula gives a measure of the conceptual density between the word and their senses. Besides, this formula has the following characteristics [14]:

- ✓ The length of the shortest path that connects the concepts involved.
- ✓ The independent measure of the number of concepts we are measuring.
- ✓ The depth in the hierarchy: concepts in a deeper part of the hierarchy should be ranker closer.

Given a concept *c*, at the top of a sub-hierarchy, and given *nhyp* and *h* (mean number of hyponyms per node and height of the sub-hierarchy, respectively), the Conceptual Density for *c* when its sub-hierarchy contains a number *m* (marks) of senses of the words to disambiguate is given by the formula below [14]:

$$CD(c,m) = \frac{\sum_{i=0}^{m-1} nhyp^{i^{0.20}}}{descendientes_c}$$

Rosso and Buscaldi [11] transform this conception of Agirre and Rigau to use the formula as an expert in the Fuzzy counting Borda system. The formulation of Conceptual Density for a sub-graph S of WordNet is defined by the following formula:

$$CD(m, f, n) = m^{\alpha}(\frac{m}{n})$$

where m is the relevant synsets in the sub-graph and n is the total number of them. The constant α takes values around 0.1, and this value was obtained through experimental results from Rosso and colleagues work [15]. The argument value f is obtained from the frequency of the synsets related in the sub-graph, which appears in WordNet. The relevant synsets match with the ones of the target word and the ones of the context words. At Buscaldi and Rosso [11] the expert that perform over Conceptual Density use as context two nouns for the word sense disambiguation process. The weights that are calculated by the previous formula are used to compute the confidence values used for fill the preference matrix. It proposes, besides, a second expert that exploits the holonymy relations instead of hypernymy. This expert uses as context all nouns in the sentence where appears the target word [11].

Buscaldi and Rosso proposition reach good precision (around 75%) over nouns and use a window of two nouns. For verbs does not behaves in a adequately way, taking into account Banerjee analysis [13] we can corroborate that to resolve correctly the verb ambiguity it is necessary use windows of at least 11 context words and besides including the major quantity of linguistic information.

3. Comparison of the Word Sense Disambiguation Algorithms

As it is known, the metrics that are used to evaluate a disambiguation algorithm are the following [12]:

Precision= # correctly disambiguated words / # disambiguated words

Recall = # correctly disambiguated words / # tested set words

Coverage = # disambiguated words / # tested set words

The combination of precision and recall it is known as F1 measure and it is calculated by the following formula [12]:

F1 = 2 * precision * recall / (precision + recall)

Table 1 shows a comparison between the not supervised word sense disambiguation algorithms according to the previous metrics [12]:

Even when Sense probability, using always the most frequently sense, is the one that occupies the first place, has a limitation, do not take into account linguistic information. Therefore, the best systems are those that use clustering and Fuzzy Borda counting. It is part of supposing then that the combination of them offer better results.

4. Word Sense Disambiguation Algorithm Propose

Supervised methods offers better solution than not supervised ones, however, the first ones need a training tagged set, this implies human factor for tagging and lacks a representative corpus for Spanish. Not supervised methods, on the other hand, do not require a training corpus, they use eternal resources as lexical data bases, thesaurus, and dictionaries, those are available in Internet and they abound most of all for the English language.

There are propositions that try to combine not supervised methods by a voting system [11,15] obtaining good results. It would be interesting combine not supervised methods under a supervised approach, that means, make a proposition where the first step is the selection of algorithm set taking into account their results in the word sense disambiguation process and then apply to them a classifier set to learn which method has to use under a determined circumstance. Previously we analyze some algorithms propositions that show satisfactory results and we conclude that ambiguity on verbs is harder to result, that is why it makes necessary to include as much linguistic information as it is possible and enlarge the sentence context to paragraph context.

The method we propose is based on the combination of various not supervised algorithms and baselines. It creates a feature vector (some features are obtained from WordNet) for each sense of the target word. Each algorithm is a feature for each sense, the output of the algorithm will be normalized in a measure between 0 and 1, then it combines certain classifiers to search the winner sense. If no exit one, then it uses a baseline (for instance MFS) in order to search the correct sense of the word.

Table 1. Not supervised system score ranked by F1 measure. (C=Coverage, P=Precision, R=Recall, F1=F1 measure)

System	C	P	R	F1
BL_{MFS}	100.0	78.89	78.89	78.89
Fuzzy Borda	100.0	78.63	78.63	78.63
Clustering based	100.0	70.21	70.21	70.21
Conceptual density	86.2	71.2	61.4	65.94
Extended Lesk	100.0	62.4	62.4	62.4

The method defines a target word as a set of sense vectors. Each vector contains in the beginning all kind of linguistic information and the normalized result of applying some algorithms (the first experiments will use Extended Lesk, Conceptual Density and Clustered based algorithms). Then the vectors are reduced by use of feature selection eliminating linguistic redundancies among vectors. To find the winner sense we suggest the use of the combination of algorithms results that we found on each feature vector, then apply a classification technique such as Support Vector Machines or Näives Bayes which performs good over two class, this classes proposed are best and bad senses. We hope this method increase the precision in word sense disambiguation process.

5. Conclusions

The supervised methods offer better solutions than not supervised ones, but they do not make use of external resources, that is why they are applied on specific domains, using language characteristics and syntactic resolution, making them language dependents.

The algorithm proposed tries to combine positive features of the not supervised method, establishing a classifier system to determine which of them is better, and taking into account the winner algorithm, the proposition output the correct sense of the word. Even when this method is in a development phase and determination of the classifiers to use, in order to validate it in some task of the SemEval 2007 competition, empirically is well suppose that this proposition shows better results taking into account the enlargement of the considered linguistic information in the algorithm previously analyzed and they are combined under a automatic learning approach.

REFERENCES

[1] E. Agirre and G. Rigau, "Word sense disambiguation using conceptual density," International Conference on Computational Linguistics (COLING), Copenhagen, Denmark 1996.

[2] H. Anaya-Sánchez, A. Pons-Porrata, *et al.*, "TKB-UO: Using sense clustering for WSD," 4th International Workshop on Semantic Evaluations (SemEval), Prague, Czech Republic, Association for Computational Linguistics, 2007.

[3] S. Banerjee, "Adapting the lesk algorithm for word sense disambiguation using wordnet," Department of Computing Science. Minnesota, USA, University of Minnesota, MSc.: 98, 2002.

[4] D. Buscaldi and P. Rosso, "UPV-WSD: Combining different WSD methods by means of fuzzy borda voting," 4th International Workshop on Semantic Evaluations (SemEval), Prague, Czech Republic, Association for Computational Linguistics, 2007.

[5] Y. Chali and S. R. Joty, "UofL: Word dense disambiguation using lexical cohesion," 4th International Workshop on Semantic Evaluations (SemEval), Prague, Czech Republic, Association for Computational Linguistics, 2007.

[6] A. Gelbukh and G. Sidorov, "Procesamiento automático del español con enfoque en recursos léxicos grandes," México, Centro de Investigación en computación, Instituto Politécnico Nacional, 2006.

[7] J. Huang, J. Lu, *et al.*, "Comparing naive Bayes, decision trees, and SVM with AUC and accuracy," Third IEEE International Conference on Data Mining, 2003.

[8] R. Ion and D. Tufis, "RACAI: Meaning affinity models," 4th International Workshop on Semantic Evaluations (SemEval), Prague, Czech Republic, Association for Computational Linguistics, 2007.

[9] D. Jurafsky and J. H. Martin, "Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition," Prentice Hall, 2000.

[10] R. Koeling and D. McCarthy, "Sussx: WSD using automatically acquired predominant senses," 4th International Workshop on Semantic Evaluations (SemEval), Prague, Czech Republic, Association for Computational Linguistics, 2007.

[11] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone," 5th Annual International Conference on Systems Documentation, Ontario, Canada, ACM, 1986.

[12] I. Nica, "El conocimiento lingüístico en la desambiguación semántica automática," España, 2006.

[13] P. Resnik, "Disambiguating noun groupings with respect to wordnet senses," Third Workshop on Very Large Corpora, Massachusetts Institute of Technology Cambridge, Massachusetts, USA, 1995.

[14] D. Yarowsky, "Word-sense disambiguation using statistical models of roget's categories trained on large corpora," 15th International Conference on Computational Linguistics (COLING), Nantes, France, Association for Computational Linguistics, 1992.

[15] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," 33th Annual Meeting of the Association for Computational Linguistics (ACL'95), Cambridge, Massachussets, 1995.

