TKB-UO: Using Sense Clustering for WSD

Henry Anaya-Sánchez[1], **Aurora Pons-Porrata**[1], **Rafael Berlanga-Llavori**[2]
[1] Center of Pattern Recognition and Data Mining, Universidad de Oriente, Cuba
[2] Computer Science, Universitat Jaume I, Spain
[1] {henry,aurora}@csd.uo.edu.cu
[2] berlanga@lsi.uji.es

Abstract

This paper describes the clustering-based approach to Word Sense Disambiguation that is followed by the TKB-UO system at SemEval-2007. The underlying disambiguation method only uses WordNet as external resource, and does not use training data. Results obtained in both Coarse-grained English all-words task (task 7) and English fine-grained all-words subtask (task 17) are presented.

1 Introduction

The TKB-UO system relies on the knowledge-driven approach to Word Sense Disambiguation (WSD) presented in (Anaya-Sánchez et al., 2006). Regarding that meaningful senses of words in a textual unit must be coherently related, our proposal uses sense clustering with the aim of determining cohesive groups of senses that reflect the connectivity of the disambiguating words.

The way this proposal uses clustering for disambiguation purposes is different from those usages reported in other works of the WSD area. For example, in (Pedersen et al., 2005) textual contexts are clustered in order to represent senses for Word Sense Discrimination. Other works like (Agirre and López, 2003), cluster fine-grained word senses into coarse-grained ones for polysemy reduction. Instead, our method clusters all possible senses corresponding to all words in a disambiguating textual unit. Thus, our system implements a novel clustering approach for the contextual disambiguation of words.

We use the lexical resource WordNet (version 2.1) as the repository of word senses, and also as the provider of sense representations. It is worth mentioning that our proposal does not require the use of training data.

2 The disambiguation algorithm

Our method starts with a clustering of all possible senses of the disambiguating words. Such a clustering tries to identify cohesive groups of word senses, which are assumed to represent the different meanings for the set of disambiguating words. Then, clusters that match the best with the context are selected via a filtering process. If the selected clusters disambiguate all words, the process is stopped and the senses belonging to the selected clusters are interpreted as the disambiguating ones. Otherwise, the clustering and filtering steps are performed again (regarding the remaining senses) until the disambiguation is achieved.

Algorithm 1 shows the general steps of our proposal for the disambiguation of a set of words W. In the algorithm, *clustering* represents the basic clustering method, *filter* is the function that selects the clusters, and T denotes the intended textual context from which words in W are disambiguated (typically, a broader bag of words than W). Next subsections describe in detail each component of the whole process.

2.1 Sense Representation

For clustering purposes, word senses are represented as topic signatures (Lin and Hovy, 2000). Thus, for each word sense s we define a vector

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 322–325,
Prague, June 2007. ©2007 Association for Computational Linguistics

Algorithm 1 Clustering-based approach for the disambiguation of the set of words W in the textual context T

Input: The finite set of words W and the textual context T.

Output: The disambiguated word senses.

Let S be the set of all senses of words in W, and $i = 0$;

repeat

$i = i + 1$

G = *clustering*$(S, \beta_0(i))$

G' = *filter*(G, W, T)

$S = \bigcup_{g \in G'} \{s | s \in g\}$

until $|S| = |W|$ or $\beta_0(i+1) = 1$

return S

$\langle t_1 : \sigma_1, \ldots, t_m : \sigma_m \rangle$, where each t_i is a WordNet term highly correlated to s with an association weight σ_i. The set of signature terms for a word sense includes all its WordNet hyponyms, its directly related terms (including coordinated terms) and their filtered and lemmatized glosses. To weight signature terms, the tf-idf statistics is used, considering each word as a collection and its senses as its of documents. Topic signatures of senses form a Vector Space Model similar to those defined in Information Retrieval Systems. In this way, they can be compared with measures such as cosine, Dice and Jaccard (Salton et al., 1975).

In (Anaya-Sánchez et al., 2006), it is shown that this kind of WordNet-based signatures outperform those Web-based ones developed by the Ixa Research Group [1] in the disambiguation of nouns.

2.2 Clustering Algorithm

Sense clustering is carried out by the Extended Star Clustering Algorithm (Gil et al., 2003), 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. The way this clustering algorithm relates word senses resembles the manner in which syntactic and discourse relation links textual elements.

[1] http://ixa.si.ehu.es/Ixa/

2.3 Filtering Process

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 measure w.r.t the textual context T.

As we represent the context T in the same vector space that the topic signatures of senses, the following function can be used to score a cluster of senses g regarding T:

$$\left(|words(g)|, \frac{\sum_i \min\{\bar{g}_i, T_i\}}{\min\{\sum_i \bar{g}_i, \sum_i T_i\}}, -\sum_{s \in g} number(s) \right)$$

where $words(g)$ denotes the set of words having senses in g, $\bar{g}$ is the centroid of g (computed as the barycenter of the cluster), and $number(s)$ is the WordNet number of sense s according to its corresponding word.

Then, we rank all clusters by using the lexicographic order of their scores w.r.t. the above function.

Once the clusters have been ranked, they are orderly processed to select clusters for covering the words in W. A cluster g is selected if it contains at least one sense of an uncovered word and other senses corresponding to covered words are included in the current selected clusters. If g does not contain any sense of uncovered words it is discarded. Otherwise, g is inserted into a queue Q. Finally, if the selected clusters do not cover W, clusters in Q adding senses of uncovered words are chosen until all words are covered.

2.4 β_0 Threshold and the Stopping Criterion

As a result of the filtering process, a set of senses for all the words in W is obtained (i.e. the union of all the selected clusters). Each word in W that has only a sense in such a set is considered disambiguated. If some word still remains ambiguous, we must refine the clustering process to get stronger cohesive clusters of senses. In this case, all the remaining senses must be clustered again but raising the β_0 threshold.

Notice that this process must be done iteratively until either all words are disambiguated or when it is impossible to raise β_0 again. Initially, β_0 is defined as:

$$\beta_0(1) = pth(90, sim(S))$$

and at the i-th iteration ($i > 1$) it is raised to:

$$\beta_0(i) = \min_{p \in \{90,95,100\}} \{\beta = pth(p, sim(S)) | \beta > \beta_0(i-1)\}$$

In these equations, S is the set of current senses, and $pth(p, sim(S))$ represents the p-th percentile value of the pairwise similarities between senses (i.e. $sim(S) = \{cos(s_i, s_j) | s_i, s_j \in S, i \neq j\} \cup \{1\}$).

2.5 A Disambiguation Example

In this subsection we illustrate the use of our proposal in the disambiguation of the content words appearing in the sentence "*The runner won the marathon*". In this example, the set of disambiguating words W includes the nouns *runner* and *marathon*, and the verb *win* (lemma of the verbal form *won*). Also, we consider that the context is the vector $T = \langle runner:1, win:1, marathon:1 \rangle$. The rest of words are not considered because they are meaningless. As we use WordNet 2.1, we regard that the correct senses for the context are $runner\#6$, $win\#1$ and $marathon\#2$.

Figure 1 graphically depicts the disambiguation process carried out by our method. The boxes in the figure represent the obtained clusters, which are sorted regarding the ranking function (scores are under the boxes).

Initially, all word senses are clustered using β_0=0.049 (the 90th-percentile of the pairwise similarities between the senses). It can be seen in the figure that the first cluster comprises the sense $runner\#6$ (the star), which is the sense refering to a trained athlete who competes in foot races, and $runner\#4$, which is the other sense of *runner* related with sports. Also, it includes the sense $win\#1$ that concerns to the victory in a race or competition, and $marathon\#2$ that refers to a footrace. It can be easily appreciated that this first cluster includes senses that cover the set of disambiguating words. Hence, it is selected by the filter and all other clusters are

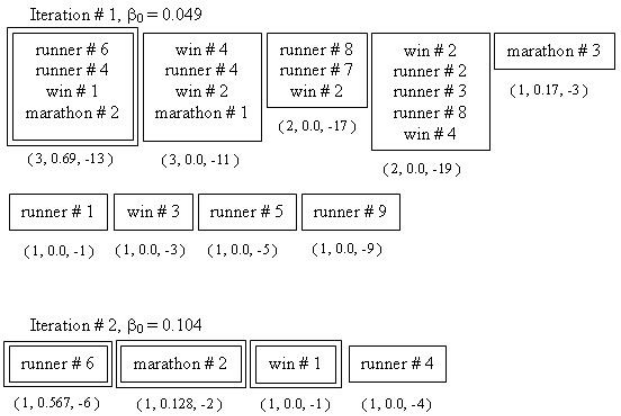

Figure 1: Disambiguation of words in "*The runner won the marathon*".

discarded. After this step, S is updated with the set $\{runner\#6, runner\#4, win\#1, marathon\#2\}$.[2]

In this point of the process, the senses of S do not disambiguate W because the noun *runner* has two senses in S. Therefore, the sttoping criterion does not hold because neither $|S| \neq |W|$ and $\beta_0(2) = 0.104 \neq 1$. Consequently, a new cluster distribution must be obtained using the current set S.

The boxes in the bottom of Figure 1 represent the new clusters. In this case, all clusters are singles. Obviously, the cluster containing the sense $runner\#4$ is discarded because the cluster that includes the sense $runner\#6$ overlaps better with the context, and therefore precedes it in the order.

Then, the final set of selected senses is $S = \{runner\#6, win\#1, marathon\#2\}$, which includes only one sense for each word in W.

3 SemEval-2007 Results

Our system participated in the Coarse-grained English all-words task (task 7) and in the English fine-grained all-words subtask (task 17). In both cases, the disambiguation process was performed at the sentence level. Thus, we defined the intended textual context T for a sentence to be the bag of all its lemmatized content words. However, W was set up in a different manner for each task.

We present our results only in terms of the F1 measure. *Recall* and *Precision* values are omitted

[2] In the figure, doubly-boxed clusters depict the selected ones by the filter.

Test set	F1
d001	0.78804
d002	0.72559
d003	0.69400
d004	0.70753
d005	0.58551
Total	0.70207

Table 1: TKB-UO results in Coarse-grained English all-words task.

Category	Instances	F1
Noun	161	0.367
Verb	304	0.303
All	465	0.325

Table 2: TKB-UO results in English Fine-grained all-words subtask.

because our method achieves a $100\,\%$ of *Coverage*.

3.1 Coarse-grained English All-words Task

Firstly, it is worth mentioning that we do not use the coarse-grained inventory provided by the competition for this task. Indeed, our approach can be viewed as a method to build such a coarse-grained inventory as it clusters tightly related senses.

Each W was defined as the set of all tagged words belonging to the sentence under consideration. Table 3.1 shows the official results obtained by our system.

As it can be appreciated, the effectiveness of our method was around the $70\,\%$, except in the fifth test document (d005), which is an excerpt of stories about Italian painters.

3.2 English Fine-grained All-words Subtask

Similar to previous task, we included into each W those tagged words of the disambiguating sentence. However, as the set of tagged words per sentence was verb-plentiful, with very few nouns, we expanded W with the rest of nouns and adjectives of the sentence.

Table 3.2 summarizes the results (split by word categories) obtained in this subtask. The second column of the table shows the number of disambiguating word occurrences.

As we can see, in this subtask only nouns and verbs were required to be disambiguated, and overall, verbs predominate over nouns. The poor performance obtained by verbs (w.r.t. nouns) can be explained by its high polysemy degree and its relatively small number of relations in WordNet.

4 Conclusions

In this paper, we have described the TKB-UO system for WSD at SemEval-2007. This knowledge-driven system relies on a novel way of using clustering in the WSD area. Also, it benefits from topic signatures built from WordNet, which in combination with the clustering algorithm overcomes the sparseness of WordNet relations for associating semantically related word senses. The system participated in both the Coarse-grained English all-words task (task 7) and the English fine-grained all-words subtask (task 17). Since we use sense clustering, we do not use the coarse-grained sense inventory provided by the competition for task 7. Further work will focus on improving the results of fine-grained WSD.

References

Eneko Agirre and Oier López. 2003. Clustering wordnet word senses. *Proceedings of the Conference on Recent Advances on Natural Language Processing*, pp. 121–130

Henry Anaya-Sánchez, Aurora Pons-Porrata, and Rafael Berlanga-Llavori. 2006. Word sense disambiguation based on word sense clustering. *Lecture Notes in Artificial Intelligence*, 4140:472–481.

Reynaldo Gil-García, José M. Badía-Contelles, and Aurora Pons-Porrata. 2003 Extended Star Clustering Algorithm. *Lecture Notes on Computer Sciences*, 2905:480–487

Chin-Yew Lin and Eduard Hovy. 2000. The Automated Acquisition of Topic Signatures for Text Summarization. *Proceedings of the COLING Conference*, pp. 495–501

Ted Pedersen, Amruta Purandare, and Anagha Kulkarni. 2005. Name Discrimination by Clustering Similar Contexts. *Lecture Notes in Computer Science*, 3406:226–237

Gerard Salton, A. Wong, and C.S. Yang. 1975. A Vector Space Model for Information Retrieval. *Journal of the American Society for Information Science*, 18(11):613–620