Alicante at CLEF 2009 Robust-WSD Task*

Javi Fernández, Rubén Izquierdo, and José M. Gómez

University of Alicante, Department of Software and Computing Systems
San Vicente del Raspeig Road, 03690 Alicante, Spain
javier.fernandez@eltallerdigital.com, ruben@dlsi.ua.es, jmgomez@ua.es
http://www.dlsi.ua.es

Abstract. In this paper we explore the use of semantic classes in an information retrieval system in order to improve the results in the Robust-WSD task at CLEF 2009. Thus, we use two different ontologies of semantic classes (WordNet domain and Basic Level Concepts) in order to re-rank the retrieved documents and obtain better recall and precision. Finally, we implement a new method to weight the expanded terms taking into account the weights of the original query terms and their relations in WordNet with respect to the new ones which have demonstrated to improve the results.

1 Introduction

The Robust-WSD task started in the last edition with two main goals: measure the robustness of the retrieval systems (good stable performance over all queries), and test the benefits of the use of Word Sense Disambiguation (WSD) on this kind of systems.

At the second edition of this task, we have decided to use an already implemented and evaluated system in the last year edition as starting point for our approach. From all the available systems, we have chosen the Universidad Complutense de Madrid system [10], because of its good results, availability and the possibility of adapting the code easily to our objectives.

Our main strategy consists of experimenting the benefits of the use of semantic classes in Information Retrieval (IR) systems. However, we propose, also, a new and flexible way of weighting terms for the query expansion based on WordNet relations.

WSD, the task of assigning the correct sense to words depending on the context in which they appear, is a hard task, difficult to be addressed and, despite the long time it has been studied, the results of state-of-art WSD systems are still a long way to be useful in other natural language processing applications, as shown in last international evaluations [15, 12]. Generally, supervised systems have obtained better results than the unsupervised ones on previous cited international

* This paper has been partially supported by the Spanish government, project TIN-2006-15265-C06-01 and by the framework of the project QALI-ME, which is a 6th Framework Research Programme of the European Union (EU), contract number: FP6-IST-033860 and by the University of Comahue under the project 04/E062.
evaluations. The annotated corpora used commonly in supervised approaches are manually tagged by lexicographers with word senses taken from a particular lexical semantic resource (most commonly WordNet [4].) WordNet has been widely criticized for being a sense repository that often provides too fine–grained sense distinctions. On the one hand, too fine–grained senses are not useful for higher level applications like machine translation or question answering. On the other hand, it seems that many word–sense distinctions are too subtle to be captured by automatic systems with the current small volumes of word–sense annotated examples. This can be a reason of the poor results of current WSD systems.

A possible solution that have been studied is the use of semantic classes instead of word senses. The task of WSD consists on assigning the proper semantic class to each ambiguous word, instead of its word sense. The use of semantic classes has several advantages. Firstly, semantic classes provide richer and more useful information than word senses. For example, for information retrieval could be more informative that the word church belongs to the semantic class building, instead of know that the correct sense to that word is the 1.

Secondly, the average polysemy of texts is decreased with the use of semantic classes due to semantic classes can group in the same class several senses of a concrete word. Therefore, the classification task is simplified. And finally, the amount of training data for each classifier is increased due to semantic classes can group together senses of different words, senses of the same word, and senses of word of different morphological categories. Thus, the number of examples to train each classifier is increased in semantic class approaches, and the problem of the lack of data in alleviated.

In [5] they empirically explored on the supervised WSD task the performance of different levels of abstraction provided by WordNet Domains [8], SUMO labels [9], Lexicographer Files of WordNet [4] and Basic Level Concepts [6]. They referred to this approach as class–based WSD since the classifiers were created at a class level instead of at a sense level. As we have said, class–based WSD cluster senses of different words into the same explicit and comprehensive grouping. Only those cases belonging to the same semantic class are grouped to train the classifier. For example, the coarser word grouping obtained in [14] only has one remaining sense for “church”. Using a set of Base Level Concepts [6], the three senses of “church” are still represented by faith.n#3, building.n#1 and religious.ceremony.n#1.

We think that IR could take advantage with the use of word sense disambiguation, but from a semantic class point of view instead of the traditional word sense point of view. Due to the data of the robust adhoc IR task has been processed automatically by two WSD systems, and the information of word senses is available, we do not run any class–based WSD system over the data.

The next section describes the architecture of our system. In section 3 we discuss the results of this system at CLEF 2009 Robust-WSD Task. Finally, in section 4 we draw the conclusions and future works.
2 Description of the System

The system architecture is shown in Figure 1.

The user query is prepared to obtain a set of terms without stopwords and any special symbol. Next, a ranked list of relevant documents are retrieved using Lucene search engine\(^1\). With the retrieved documents, the initial query, the relations of the external resource WordNet and state-of-art query expansions methods an expanded query is obtained. The terms of this new query are weighted taking into account the weights of the original query terms, their relations in WordNet with respect to the new ones, the weight assigned by the WSD system to each sense and the weight returned by the expansion method. Once we have a new list of weighted terms, we do, again, a search but using the expanded query instead of the original one in order to retrieved a new ranked list of documents. Finally, we use the semantic class information from two different semantic resources (WordNet Domains and Base Level Concepts) in order to obtain a re-ranked document list as result.

In following sections we are explaining each of this processes in more detail.

\(^1\) http://lucene.apache.org
2.1 Search engine and query expansion

The search engine, which we are using, is the one provided by the Universidad Complutense de Madrid as we have already mentioned above. Their implementation is a modified version of Lucene which uses the BM25 probabilistic model [13] for the document retrieval. They have also implemented two state-of-art query expansion methods: Kullback-Liebler Divergence [2, 3] (an information-theoretic approach) and the Bo1 model [7, 11] (based on Divergence From Randomness [1]). For our system we have chosen the Bo1 model because it is the approach with the best results in their evaluations. We also have decided to use the same constant values than they used in the last CLEF Robust-WSD edition in order to compare the effectiveness of our methods of semantic classes.

As we can see in the Figure 1, we make two search processes. For the first retrieval process, query terms are lemmatized and stemmed in order to increase the system recall. The first Search module gets these terms as input and returns a list of relevant documents using the BM25 probabilistic model. Next, in the Query expansion module, we expand the original query obtaining new terms by means of the Bo1 model.

Although [10] proposed a method for weighting the expanded query terms based on WordNet, we have preferred to use our own method due to the fact that they do not use all senses of each term but the weightest one. In our system we have decided to use all senses retrieved by the WSD system in order to improve the recall. In this way, the system searches all expanded terms in the relations of WordNet with respect to the synonyms, hyperonyms and hyponyms until a certain level or distance. For example, if the distance was 2, we search any expanded term among the hyperonym and hyponym synsets of the original terms but, also, the hyperonyms of the hyperonyms and the hyponyms of the hyponyms. The distance constant marks the allowed jumps to reach in the WordNet relations from the synsets of the query terms. As we have mentioned above, we use all senses supplied for the WSD system for each query term but we take into account the score given by these systems to each sense in order to calculate the weight of the expanded terms. Therefore, this distance factor is calculated by the following equation:

\[
weight(synset_{i,d}) = weight(synset_{i,d-1}) \times \alpha^d
\] (1)

We defined \(synset_{i,1}\) as a WordNet synset given and \(synset_{i,d}\) as another WordNet synset which is related to the \(synset_{i,1}\) of a distance of \(d\) jumps (taken into account only hyperonym and hyponym relations). Thus, \(weight(synset_{i,d})\) is the weight of the synset \(i, d\) and \(weight(synset_{i,1})\) is the score given by the WSD system to the synset \(i, d\). \(\alpha\) is a constant whose value is between 0 and 1 and \(d\) the distance of \(synset_{i,d}\) to the \(synset_{i,1}\).

Once we have calculated the previous synset weight, we combine this weight with the weight assigned by the expanded method Bo1 in order to calculate the final term weight using the following equation:
weight(term_t) = \frac{weight(synset_{i,d}) + weight_0(term_t)}{2} \tag{2}

Where weight(term_t) is the weight of the expanded term t which is grouped in the WordNet synset i, d, and weight_0(term_t) is the weight assigned by bo1 to the term t.

With these equations, we give importance to those expanded terms which are more related with the original query terms and, in addition, we include the score given by the WSD system for each query term in the final term weight. Thus, we include all senses of a term in the search but we give more importance those terms which are related with more likely senses and closer to the original query terms.

2.2 Semantic classes

Our approach consists on mapping the assigned word senses to semantic classes, specifically to WordNet Domains labels and Basic Level Concepts.

WordNet Domains [8] is a hierarchy of 165 Domain Labels which have been used to label all the WordNet synsets. Information brought by Domain Labels is complementary to what is already in WordNet. First of all, a Domain Label can include synsets of different syntactic categories: for instance MEDICINE groups together senses from nouns, such as doctor or hospital, and from verbs, such as to operate. Second, a Domain Label may also contains senses from different WordNet subhierarchies. For example, SPORT contains senses such as athlete, deriving from life form, game equipment from physical object, sport from act and playing field from location.

Basic Level Concepts [6] are a set of concepts that result from the compromise between two conflicting principles of characterization:

- Represent as many concepts as possible;
- Represent as many features as possible;

As a result of this, Basic Level Concepts typically occurs in the middle of hierarchies and less than the maximum number of relations.

The authors developed a method for the automatic selection of BLC from WordNet. They use a very simple method for deriving a small set of appropriate meanings using basic structural properties of WordNet. The approach considers:

- The total number of relations of every synset or just the hyponymy relations
- Discard those BLCs that do not represent at least a number of synsets.
- Optionally, the frequency of the synsets (summing up the frequency of the senses provided by WordNet).

The process of automatic selection of BLC follows a bottom-up approach using the chain of hypernym relations. For each synset in WN, the process selects as its Base Level Concept the first local maximum according to the relative number of relations. For synsets having multiple hypernyms, the path having
the local maximum with higher number of relations is selected. Usually, this process finishes having a number of fake Base Level Concepts. That is, synsets having no descendants (or with a very small number) but being the first local maximum according to the number of relations considered. Thus, the process finishes checking if the number of concepts subsumed by the preliminary list of BLC is higher than a certain threshold. For those BLC not representing enough concepts according to a certain threshold, the process selects the next local maximum following the hypernym hierarchy.

Thus, depending on the type of relations considered to be counted and the threshold established, different sets of BLC can be easily obtained for each WN version. For our work, we have selected the set of BLC built using all kind of relations and a threshold of 20 as the minimum number of synsets that each BLC must subsume.

We explain now the representation of documents or queries with semantic classes of the words contained on them. In the task data, each ambiguous word is annotated with their possible senses, each one with a certain probability. Starting from this information, we create a domain vector, for a query or for a document, containing all the semantic classes information of the query or document. The Domain vector consists on a vector which, each element, representing a WordNet Domain or a Basic Level Concept and its associated weight. Note that there are 165 Domain Labels in WordNet Domains and 558 Basic Level Concepts for nouns. The way to build this vector is: each word has annotated several senses, with the associated probability; each word sense is mapped to its proper semantic class, and the element of the vector corresponding to this Domain is increased with the probability associated to the word sense. When all the words are processed, we obtain a domain vector representing the semantic information of the document or query. Finally to compare two documents, or a document and a query, and obtain their similarity in terms of their semantic content, we use the value of the cosine defined by the two domain vectors.

2.3 Integration of Semantic classes in Robust Ad hoc

Once the final list of documents from the expanded query is retrieved, the Semantic re-ranking module re-arrange this list taking into account both the similarity returned by the BM25 probabilistic model and the similarity calculated by semantic class system. In order to do this, we have studied the following 5 different equations:
\[
\text{semsim}_1(i, j) = \text{sim}_{ij} \times \text{sem}_{ij}
\]  
(3)

\[
\text{semsim}_2(i, j) = \begin{cases} 
\text{sim}_{max} + \text{sim}_{ij} & \text{if } \text{sem}_{ij} > h \\
\text{sim}_{ij} & \text{otherwise}
\end{cases}
\]  
(4)

\[
\text{semsim}_3(i, j) = \begin{cases} 
(\text{sim}_{max} + \text{sim}_{ij}) \times \text{sem}_{ij} & \text{if } \text{sem}_{ij} > h \\
\text{sim}_{ij} & \text{otherwise}
\end{cases}
\]  
(5)

\[
\text{semsim}_4(i, j) = \begin{cases} 
\text{sim}_{max} + \text{sim}_{ij} \times \text{sem}_{ij} & \text{if } \text{sem}_{ij} > h \\
\text{sim}_{ij} & \text{otherwise}
\end{cases}
\]  
(6)

\[
\text{semsim}_5(i, j) = \text{sim}_{max} \times \text{sim}_{ij} + \text{sim}_{ij}
\]  
(7)

Where \( \text{semsim}_x(i, j) \) is the final similarity between the query \( i \) and the document \( j \) using the method \( x \), \( \text{sim}_{ij} \) is the similarity of the query \( i \) with respect to the document \( j \) returned by the search engine, \( \text{sem}_{ij} \) is the same similarity but returned by the semantic class system, \( \text{sim}_{max} \) is the greatest value of similarity returned by the search engine for the query \( i \) and \( h \) is a constant which determines a semantic similarity threshold defined empirically.

As both similarity values (\( \text{sim}_{ij} \) and \( \text{sem}_{ij} \)) are normalized ones, our first approximation was the equation 3. In this equation we simply multiply both values (the similarity returned by the search engine and the similarity obtained by the semantic class system) in order to obtain a new similarity value between the query \( i \) and the document \( j \). With this equation both values have the same importance.

The equation 4 was thought in order to put those documents with a certain level of semantic relation with the query above other ones. Therefore, if the similarity obtained by the search engine will be summed to the greatest similarity if, and only if, the semantic class similarity between the document and the query is greater than a threshold \( h \), otherwise only the BM25 similarity will be taken into account.

The next equation (5) is based on the previous one but the semantic similarity multiplies the sum of the BM25 similarity for the document \( j \) and the greatest similarity. The idea of this equation is to overlap some of the less semantic related documents which exceed the threshold with those which do not exceed.

In order to give more relevance those documents with high semantic similarity but taking into account the semantic class score in the final similarity value, the equation 6 was used.

Finally, the equation 7 multiplies the greatest similarity value with the document semantic similarity and, next, sums this result to the search engine document similarity. This equation tries to improve the search engine similarity value using the semantic similarity as reduction factor of maximum similarity value.
3 Evaluation

In this section we report the results of each one of our proposals separately.

For the evaluation of the Expanded query terms weighter, we have to set the value for two variables: $\alpha$ and $d$ (distance). In order to get the best values for these variables, we have experimented with several different values for them. In the Table 1 we present two of the best results of those experiments. With $\alpha = 0.8$ and $d = 1$ we improve the baseline GMAP in a 9.97%. With $\alpha = 0.92$ and $d = 6$ we improve both the baseline MAP in a 0.02% and the baseline GMAP in a 8.19%.

Table 1. Evaluation of the Expanded query terms weighter module

<table>
<thead>
<tr>
<th></th>
<th>MAP</th>
<th>GMAP</th>
<th>R-Prec</th>
<th>P@5</th>
<th>P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25 + Bo1 (Baseline)</td>
<td>.3737</td>
<td>.1294</td>
<td>.3585</td>
<td>.4475</td>
<td>.3825</td>
</tr>
<tr>
<td>BM25 + Bo1 + WD ($\alpha = 0.8, d = 1$)</td>
<td>.3706</td>
<td>.1423</td>
<td>.3624</td>
<td>.4500</td>
<td>.3750</td>
</tr>
<tr>
<td>BM25 + Bo1 + WD ($\alpha = 0.92, d = 6$)</td>
<td><strong>.3738</strong></td>
<td>.1400</td>
<td><strong>.3655</strong></td>
<td><strong>.4513</strong></td>
<td>.3775</td>
</tr>
</tbody>
</table>

The results of the second of our proposals, the Semantic re-ranking module, depend on the function used for the integration of the documents’ weights given by the semantic classes and the search engine. In addition, for some of the integration functions, the variable $h$ (threshold) has to be set too. We have experimented with those functions and different values for the threshold. For each integration function we have obtained its best threshold value (when needed), as can be seen in Tables 2 and 3.

Table 2. Evaluation of the Semantic re-ranking module with WND and different integration functions

<table>
<thead>
<tr>
<th></th>
<th>MAP</th>
<th>GMAP</th>
<th>R-Prec</th>
<th>P@5</th>
<th>P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25 + Bo1 + WND + RR1</td>
<td><strong>.3752</strong></td>
<td>.1298</td>
<td><strong>.3638</strong></td>
<td><strong>.4462</strong></td>
<td><strong>.3862</strong></td>
</tr>
<tr>
<td>BM25 + Bo1 + WND + RR2 ($h = 0.2$)</td>
<td>.3737</td>
<td>.1294</td>
<td>.3585</td>
<td>.4475</td>
<td>.3825</td>
</tr>
<tr>
<td>BM25 + Bo1 + WND + RR3 ($h = 1.0$)</td>
<td>.3737</td>
<td>.1294</td>
<td>.3585</td>
<td>.4475</td>
<td>.3825</td>
</tr>
<tr>
<td>BM25 + Bo1 + WND + RR4 ($h = 0.5$)</td>
<td><strong>.3752</strong></td>
<td>.1298</td>
<td><strong>.3638</strong></td>
<td><strong>.4462</strong></td>
<td><strong>.3862</strong></td>
</tr>
<tr>
<td>BM25 + Bo1 + WND + RR5</td>
<td>.3746</td>
<td>.1296</td>
<td>.3592</td>
<td>.4463</td>
<td>.3856</td>
</tr>
</tbody>
</table>

As we can see in Table 2, some of the results of different integration functions are the same. In the case of RR2, RR3 and RR5, this occurs because the best results they can reach are the same as the baseline results. This integration functions do not improve the system. In the case of RR1 and RR4, they mathematical function are the same, except the second one, that can be affected by
Table 3. Evaluation of the Semantic re-ranking module with BLC20 and different integration functions

<table>
<thead>
<tr>
<th></th>
<th>MAP</th>
<th>GMAP</th>
<th>R-Prec</th>
<th>P@5</th>
<th>P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25 + Bo1 + BLC20 + RR1</td>
<td>.3533</td>
<td>.1231</td>
<td>.3375</td>
<td>.4337</td>
<td>.3619</td>
</tr>
<tr>
<td>BM25 + Bo1 + BLC20 + RR2 (h = 0.8)</td>
<td>.3776</td>
<td>.1317</td>
<td>.3609</td>
<td>.4437</td>
<td>.3806</td>
</tr>
<tr>
<td>BM25 + Bo1 + BLC20 + RR3 (h = 0.8)</td>
<td>.3776</td>
<td>.1317</td>
<td>.3609</td>
<td>.4437</td>
<td>.3806</td>
</tr>
<tr>
<td>BM25 + Bo1 + BLC20 + RR4 (h = 0.8)</td>
<td>.3776</td>
<td>.1317</td>
<td>.3609</td>
<td>.4437</td>
<td>.3806</td>
</tr>
<tr>
<td>BM25 + Bo1 + BLC20 + RR5</td>
<td>.3375</td>
<td>.1170</td>
<td>.3229</td>
<td>.4213</td>
<td>.3625</td>
</tr>
</tbody>
</table>

the threshold. Both of them are have the best results for the WND model for all measures.

In Table 3, we show the results of the integration with the BLC20 model. The best results are obtained by the RR2, RR3 and RR4 integration functions. These are the functions that use a threshold. Note the threshold that gives the best results is the same for all them. This can make us think that the threshold \( h = 0.8 \) is the ideal for determining if a semantic class is relevant or not.

For our final comparison, we have chosen the best integration function for each model (WND, BLC20), as shown on Table 4 and Figure 2.

Fig. 2. Results of MAP and GMAP for models WND and BLC20 and each integration function

The integration of the semantic classes to the search engine improves the baseline results. With WND we improve both the baseline MAP in a 0.4% and the baseline GMAP in a 0.31%. With BLC20 we improve both the baseline MAP in a 0.64% and the baseline GMAP in a 1.77%.
### Table 4. Evaluation of the Semantic re-ranking module

<table>
<thead>
<tr>
<th></th>
<th>MAP</th>
<th>GMAP</th>
<th>R-Prec</th>
<th>P@5</th>
<th>P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25 + Bo1 (Baseline)</td>
<td>.3737</td>
<td>.1294</td>
<td>.3585</td>
<td>.4475</td>
<td>.3825</td>
</tr>
<tr>
<td>BM25 + Bo1 + WND</td>
<td>.3752</td>
<td>.1298</td>
<td>.3638</td>
<td>.4462</td>
<td>.3862</td>
</tr>
<tr>
<td>BM25 + Bo1 + BLC20</td>
<td>.3776</td>
<td>.1317</td>
<td>.3609</td>
<td>.4437</td>
<td>.3806</td>
</tr>
</tbody>
</table>

4 Conclusions

The results of the experiments with our two proposals have shown improvements to the initial information retrieval system.

In the first one, the Expanded query terms weighter module, we have experimented with the weights of the terms in a probabilistic IR system. We have applied a smoothing function based on the WordNet distance to the weights given by the IR system. The experiments made have shown GMAP improvements of nearly 10% but not significant MAP improvements.

As future work we propose to continue with the experiments on this module. For the propagation function 2, the search of the best values for $\alpha$ and $d$ can be more exhaustive, finding better values for this variables. Moreover, new relations can be explored in WordNet (not only hyponyms and hyperonyms), in order to improve recall. Even new weight propagation functions can be proposed to better exploit the concept of distance in WordNet.

In the second of our proposals, the Semantic re-ranking module, we have integrated the semantic classes to a IR system. We have done this integration recalculating the weight of the documents retrieved depending on the similarity between the semantic class of each document and the semantic class of the query. The results of the experiments made reveal that the semantic classes resources can be effectively be integrated to the IR systems.

This module can also be led to new levels. We only have used five simple integration functions for the search engine and the semantic classes weights. More functions can be studied to find the best way to integrate the available resources of semantic classes.

References


