Using Sense Clustering for the Disambiguation of Words

Henry Anaya-Sánchez, Aurora Pons-Porrata, and Rafael Berlanga-Llavori

***Abstract*—Clustering methods have been extensively used in the solution of many Information Processing tasks in order to capture unknown object categories. This paper presents an approach to Word Sense Disambiguation based on clustering. The underlying idea is that the clustering of word senses provides a useful way to discover semantically related senses. We evaluate our proposal regarding both fine- and coarse-grained disambiguation. Experimental results over Senseval-3 all-words, SemCor 2.0 and SemEval-2007 corpora are presented. Promising values of precision and recall are obtained.**

***Index Terms*—Word sense disambiguation, clustering.**

I. Introduction

THE task of Word Sense Disambiguation (WSD) consists of selecting the appropriate sense for a particular contextual occurrence of a polysemous word. This task can be specialized according to the sense definitions. For instance, word sense induction refers to the process of discovering different senses of an ambiguous word without prior information about the inventory of senses [21]. On the other hand, there are two major approaches for the disambiguation when predetermined sense definitions are provided: data-driven (or corpus-based) and knowledge-driven WSD. Data-driven methods are supervised because they require a learning model built from hand-tagged samples to disambiguate words. Instead, knowledge-driven methods exploit word relationships provided by a background knowledge source, avoiding thus the use of samples. Currently, lexical resources like WordNet [14] constitute the referred source in most cases.

WSD can be seen as a categorization problem consisting of assigning a category label (predefined sense) to each word. In this way, data-driven approaches can be regarded as supervised categorization methods, whereas knowledge-driven ones as unsupervised.

Clustering is one of the most accepted unsupervised categorization methods. It has been explicitly used in WSD for two main purposes. The first one consists of clustering textual contexts to represent different senses in corpus-driven WSD (e.g. [17]) and to induce word senses (e.g. [18], [3]). The other purpose has been the clustering of fine-grained word senses into coarse-grained ones for reducing the polysemy degree of words (e.g. [13], [1]). However, clustering has not been used as categorization method for WSD, that is, as a way to identify sets of word senses that are semantically related.

In this paper, we present a knowledge-driven approach to WSD based on sense clustering. Basically, our proposal uses sense clustering to capture the reflected cohesion among the words of a textual unit. More specifically, starting from an initial clustering of all the possible senses for a textual unit, clusters of senses with a high cohesion w.r.t the textual context are selected. The senses belonging to the selected clusters are grouped and selected again until all words are disambiguated.

The rest of the paper is organized as follows. First, Section II presents our proposal for the disambiguation of words. Section III describes some experiments carried out over Senseval-3 all-words, SemCor 2.0 and SemEval coarse-grained corpora. Finally, Section IV is devoted to offer some considerations and future work as conclusions.

II. Word sense clustering

In this section we address the problem of disambiguating a finite set of words $W = \{w_1, \ldots, w_n\}$ w.r.t its textual context T. The underlying idea of sense clustering is that meaningful word senses must be associated by means of a certain complex relation, which is non-relevant for our purposes because we are only interested in the senses it links. Hence, we propose to identify cohesive groups of senses which are assumed to represent different meanings for the set of words W. Finally, those clusters that fit in with the context T contain the suitable senses.

Algorithm 1 shows the general steps of our proposal. In the algorithm, *clustering* represents the basic clustering algorithm which groups word senses and, *filter* denotes the filtering process which selects the clusters that allow the disambiguation of words in W. The filtering process is described in Algorithm 2. Next paragraphs describe in detail the whole process.

a) Topic signatures: In our approach word senses are represented as topic signatures [12]. Thus, for each word sense s we define a vector $\langle t_1 : \sigma_1, \ldots, t_m : \sigma_m \rangle$, 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.

Manuscript received November 4, 2008. Manuscript accepted for publication August 28, 2009.

Henry Anaya-Sánchez and Aurora Pons-Porrata are with Center for Pattern Recognition and Data Mining, Universidad de Oriente, Santiago de Cuba, Cuba (henry@cepramid.co.cu, aurora@cepramid.co.cu).

Rafael Berlanga-Llavori is with Department of Langauges and Computer Systems, Universitat Jaume I, Castelló, Spain (berlanga@lsi.uji.es).

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

Algorithm 2 Definition of the filtering process

Input: The set of clusters G, the finite set of words W and the textual context T.
Output: The set of selected clusters G'.
for all g in G **do**
$scores(g) = compare(g, T)$
end for
Sort all groups in G by using the lexicographic order of its scores
Let Q be an empty queue, and G' an empty set
for all g in G **do**
if $\exists(s \in g) \forall(g' \in G')[words(\{s\}) \cap words(g') = \emptyset \quad \wedge$
$\forall(s' \in g)[words(\{s'\}) \subseteq words(g') \implies s' \in \bigcup_{g'' \in G'} g'']]$
then
G'=$G' \cup \{g\}$
else if $\neg\exists(s \in g) \forall(g' \in G')[words(\{s\}) \cap words(g') = \emptyset]$
then
Discard g
else
$Q.insert(g)$
end if
end for
while $words(\bigcup_{g' \in G'} g') \neq W$ **do**
g=$Q.front\_element$
G'=$G' \cup \{g\}$
$Q.remove\_front\_element()$
end while
return G'

To weight signature terms, the tf-idf statistics is used, considering each word as a collection and its senses as its documents. Notice that topic signatures form a Vector Space Model similar to those defined in Information Retrieval Systems. In this way, topic signatures can be compared with usual Information Retrieval measures such as cosine, Dice and Jaccard [19].

b) Clustering algorithm: Clustering is carried out by using the Extended Star Clustering Algorithm [7], 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 relations link textual elements.

c) Cluster filtering: Once clustering is performed over all possible word senses from 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 intended textual context T. This process can be seen as a context-driven filtering of word senses.

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:

$$compare(g, T) = \left(|words(g)|, \frac{\sum_i \min(\bar{g}_i, T_i)}{\min(\sum_i \bar{g}_i, \sum_i T_i)}, -\sum_{s \in g} \text{nth}(s) \right)$$

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 $\text{nth}(s)$ is the WordNet number of the sense s according to its corresponding word.

This function scores each cluster considering three measures: the number of words it has associated, its overlapping w.r.t the context and the WordNet sense frequency of its senses respectively. Therefore, we rank all clusters by using the lexicographic order of their scores w.r.t. this 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.

d) Disambiguation process: 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 only has 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 senses obtained in the previous step 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 not possible to raise β_0 no more. The following equation states how β_0 is set up at each iteration (i-th iteration):

$$\beta_0(i) = \begin{cases} \text{pth}(90, sim(S)) & \text{if } i = 1, \\ \min\limits_{q \in \{90,95,100\}} \{\beta = \text{pth}(q, sim(S)) | \beta > \beta_0(i-1)\} & \text{otherwise.} \end{cases}$$

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

runner # 1 = {<criminal,1.056>, <outlaw,1.055>, <ilegal,1.006>, <contrabandist,1.006>, ...}
runner # 2 = {<travel,1.056>, <carrier,0.930>, <arrive,0.930>, <distant,0.772>, <tourist,0.772>, ...}
runner # 3 = {<deliver,1.037>, <boy,1.006>, <announce,0.936>, <dispatch,0.772>, <message,0.718>, ...}
runner # 4 = {<bat,1.055>, <pitcher,1.037>, <base_runner,1.006>, <hit,0.930>, <manager,0.772>, ...}
runner # 5 = {<plant,1.056>, <fungus,1.005>, <structure,1.054>, <branch,1.037>, <foliage,0.930>, ...}
runner # 6 = {<race,1.056>, <olympic,1.049>, <trained,1.037>, <marathon,0.930>, <gold,0.772>, ...}
runner # 7 = {<carpet,1.056>, <covering,1.055>, <include,0.930>, <color,0.930>, <thick,0.930>, ...}
runner # 8 = {<device,1.056>, <light,1.055>, <instrument,1.055>, <metal,1.055>, <machine,1.037>, ...}
runner # 9 = {<atlantic,1.049>, <western,1.049>, <cape,1.006>, <vertebrate,1.006>, <tropical,1.006>, ...}

win # 1 = {<contest,0.654>, <gold,0.587>, <medal,0.587>, <contend,0.487>, <contestant,0.487>, ...}
win # 2 = {<acquire,0.66>, <receive,0.665>, <earn,0.662>, <possession,0.662>, <get,0.635>, ...}
win # 3 = {<score,0.587>, <advance,0.587>, <gain_ground,0.587>, <get_ahead,0.587>, ...}
win # 4 = {<goal,0.662>, <attempt,0.654>, <achieve,0.635>, <attain,0.635>, <reach,0.635>, ...}

marathon # 1 = {<task,0.518>, <endurance_contest,0.503>, <arduous,0.503>, <labor,0.465>, ...}
marathon # 2 = {<race,0.528>, <footrace,0.528>, <mile,0.503>, <yard,0.503>, <steeplechase,0.386>, ...}
marathon # 3 = {<battle,0.528>, <defeat,0.528>, <force,0.528>, <army,0.528>, <troop,0.528>, ...}

Fig. 1. Portion of the representation of senses.

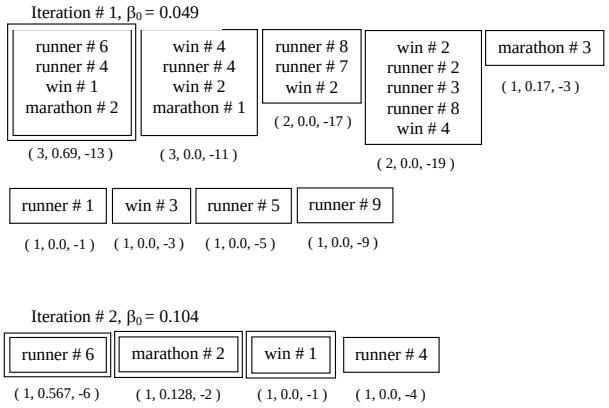

Fig. 2. Disambiguation of words in "*The runner won the marathon*".

A. An 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, in this case we consider that the context T is defined as the vector representation of the filtered and lemmatized sentence, i.e. $T = \langle runner : 1, win : 1, marathon : 1\rangle$. The rest of words are not considered because they are meaningless. As we use WordNet 2.0, we regard that the correct senses for the context are $runner\#6$, $win\#1$ and $marathon\#2$. In Figure 1, an extract of the representation of all word senses is shown.

Figure 2 graphically depicts the disambiguation process carried out by our method in the disambiguation of word senses. The boxes in the figure represent the obtained clusters, which are sorted regarding the lexicographic order given by the function $compare$ (scores are under the boxes).

Initially, the set of all word senses is clustered using the initial β_0=0.0498 (the 90th-percentile of the pairwise similarities between the senses). It can be seen 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 the sports. Also, it includes the sense $win\#1$ that concerns 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 discarded. After this step, S is updated with the set $\{runner\#6, runner\#4, win\#1, marathon\#2\}$. [1]

In this point of the process, the senses of S do not disambiguate W because the noun *runner* has two senses in S. Also, the next value for the threshold is $\beta_0(2) = 0.1043$. Therefore, the disambiguation of words does not hold because neither $|S| = |W|$ nor $\beta_0(i+1) = 1$. Consequently, a new cluster distribution must be obtained using the current set S.

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

The set of boxes in the bottom of Figure 2 represents 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 T, and therefore precedes him in the order.

Then, the set of current senses becomes $S = \{ runner\#6, win\#1, marathon\#2\}$, which includes only one sense for each word in W, and thereby the disambiguation holds and the process is stopped. Finally, the current set S is returned as the set of senses that disambiguates the verb *win*, and the nouns *runner* and *marathon*.

III. Experimental results

In order to evaluate our approach, we consider the disambiguation at two different levels of sense granularity. A fine-grained disambiguation was evaluated by using both a subset of SemCor 2.0 composed by all the documents of *brown1* and *brown2*, and a version of Senseval-3 all-words corpus (annotated with WordNet 2.0). In contrast, we use the corpus provided by Task 7 of SemEval-2007 [16] to evaluate the performance of our approach in a coarse-grained WSD.

As evaluation measures, we use the well-known *Precision*, *Recall* and *Coverage*. In the fine-grained case we use their respective "Without U" versions (defined as in Senseval-3 [20]), because there are some word senses in the corpora that are not covered by WordNet 2.0.

In both cases, the disambiguation is performed at the sentence level, i.e., we assume that there is just one correct meaning per word in each sentence. Also, each context T is defined as the vector representation (regarding all lemmatized words) of the sentence.

A. Fine-grained WSD

In this case, we carry out two kinds of experiments. In the first one, we disambiguate all words of each sentence (i.e., W is the set of all meaningful words of the sentence), whereas in the second one we only disambiguate nouns (the set W only

TABLE I
WSD PERFORMANCE OVER THE SENSEVAL-3 ALL-WORDS CORPUS.

Experiment	Category	Instances	Untagged	Precision	Recall	Coverage
	Noun	951	25	0.475	0.462	97.3 %
	Verb	751	3	0.285	0.284	99.6 %
All-words	Adjective	364	11	0.610	0.592	96.9 %
	Adverb	15	0	0.933	0.933	100 %
	All	2081	39	0.432	0.424	98.1 %
Only nouns	Noun	951	25	0.490	0.477	97.3 %

TABLE II
WSD PERFORMANCE OVER THE SEMCOR 2.0 CORPUS.

Experiment	Category	Instances	Untagged	Precision	Recall	Coverage
	Noun	88058	105	0.536	0.535	99.8 %
	Verb	48328	154	0.291	0.290	99.6 %
All-words	Adjective	35664	408	0.626	0.619	98.8 %
	Adverb	20589	837	0.623	0.598	95.9 %
	All	192639	1504	0.500	0.496	99.2 %
Only nouns	Noun	88058	105	0.542	0.541	99.8 %

contains the nouns of the sentence). We will refer to these kind of experiments as "All-words" and "Only nouns" respectively.

Table I summarizes the results obtained over the Senseval-3 all-words corpus. The third column contains the total number of disambiguating word occurrences, and the fourth column shows the number of untagged word occurrences in the corpus, i.e. word occurrences that do not have a WordNet 2.0 sense.

It is worth mentioning that the official Senseval-3 results (reported in [20]) are obtained using a version of Senseval-3 all-words corpus that has been annotated with WordNet 1.7.1. Therefore, our results can not be directly compared with them. However, unlike most participants in Senseval-3 contest, our method obtains a 100 % of coverage if untagged words are ignored.

As we can see, the best performance is obtained in the disambiguation of adverbs and adjectives, while the worst is achieved by the verbs. It can be explained by the high polysemy degree of verbs and its relatively small number of relations in WordNet. Also, it can be appreciated that disambiguating only nouns produces slightly better results than disambiguating nouns together with other words.

The results obtained by our method over the SemCor 2.0 corpus are summarized in Table II. As we can see, they are in agreement with those obtained for the Senseval-3 corpus.

In order to have a better understanding of the behaviour of the algorithm over different knowledge domains, Table III summarizes the overall precision, recall and coverage split according to the SemCor categories.

As shown in Table III, our algorithm performs the best in *Press: reportage* category. In all other categories the recall values are similar. Thus, it seems that the performance is not affected with different knowledge domains.

Finally, we compare our method with four knowledge-driven WSD algorithms: Conceptual density [2], UNED method [6], the Lesk method [11] and the Specification marks with voting heuristics [15]. Table IV includes the recall values obtained over the whole SemCor corpus considering only polysemous nouns.

TABLE III
"ALL WORDS" WSD PERFORMANCE OVER THE SEMCOR CATEGORIES.

Categories	Precision	Recall	Coverage
A. Press: reportage	0.554	0.551	99.4 %
B. Press: editorial	0.520	0.518	99.5 %
C. Press: reportage	0.508	0.505	99.3 %
D. Religion	0.492	0.491	99.7 %
E. Skill & Hobbies	0.499	0.496	99.4 %
F. Popular lore	0.510	0.507	99.3 %
G. Belles letters, biography, essays	0.489	0.487	99.6 %
H. Miscellaneous	0.528	0.525	99.4 %
J. Learned	0.513	0.511	99.6 %
K. General fiction	0.472	0.468	99.0 %
L. Mystery & detective fiction	0.498	0.489	98.1 %
M. Science fiction	0.500	0.495	98.9 %
N. Adventure & western fiction	0.470	0.462	98.3 %
P. Romance & love story	0.461	0.451	97.8 %
R. Humor	0.497	0.490	98.5 %
Brown 1	0.502	0.499	99.3 %
Brown 2	0.497	0.493	99.0 %
Whole SemCor	0.500	0.496	99.2 %

TABLE IV
COMPARISON WITH OTHER METHODS OVER SEMCOR CORPUS.

WSD method	Recall
Conceptual density	0.220
Lesk	0.274
UNED method	0.313
Specification marks	0.391
Our method using SemCor 1.6	**0.472**
Our method using SemCor 2.0	**0.426**

In this case, we experiment with two versions of the SemCor corpus: SemCor 1.6 and SemCor 2.0, and obviously with their corresponding versions of WordNet. It is due to two reasons. The first one is that the results of the other algorithms are obtained using SemCor 1.6. The other reason consists of showing the impact in the disambiguation of the higher polysemy degree of WordNet 2.0 w.r.t. WordNet 1.6. As it can be appreciated, our approach improves all other methods considering both versions of WordNet.

B. Coarse-grained WSD

As the sense inventory corresponding to the coarse-grained English all-words task of SemEval-2007 consists of clusters of WordNet 2.1 senses, we proceed in the same way as with the fine-grained case. That is, we disambiguate each set of words from a sentence w.r.t. WordNet 2.1. However, we use the coarse-grained score provided by the task organizers to evaluate our approach.

In Table V, we show the performance of our method in the coarse-grained English all-word tast of SemEval-2007. In this table we have ommited the values of *Precision* and *Coverage* because all words are disambiguated by the algorithm, i.e. *Precision* values coincide with *Recall* and a 100% of *Coverage* is achieved.

TABLE V
WSD PERFORMANCE IN TASK 7 OF SEMEVAL-2007.

Word Category	Instances	Recall
Noun	1108	0.708
Verb	591	0.626
Adjective	362	0.787
Adverb	208	0.740
All	2269	0.702

TABLE VI
OVERALL COARSE-GRAINED PERFORMANCE.

System	F1
UPV-WSD [4]	0.786
Our method	0.702
RACAI-SYNWSD [9]	0.657
SUSSX-FR [10]	0.604
UOFL [5]	0.506
SUSSX-C-WD [10]	0.459
SUSSX-CR [10]	0.457
MFS *baseline*	0.788

As it can be appreciated, like in the fine-grained experiments the category of verbs significantly perform the worst. Also, the other word categories increase their scores w.r.t the fine grained case because of the relaxation of this new task.

In order to contextualize our results in the current State-of-the-Art, we show in Table VI a comparison between our results and those obtained by other unsupervised systems that participated in SemEval-2007 along with the Most Frequent Sense (MFS) baseline. Systems are ranked according to their F1 score (harmonic mean between *Precision* and *Recall*).

As it can be appreciated, our method obtains the second highest score, which constitutes a good result. It is worth mentioning that unlike most other methods, our proposal does not use any external resource except WordNet, neither the coarse-grained sense inventory provided by the task organizers. Also, it is not used the MFS backoff strategy.

IV. CONCLUSION

In this paper a new approach for the disambiguation of words has been proposed. Its novelty relies on the use of clustering as a natural way to connect semantically related word senses.

Most existing approaches attempt to disambiguate a target word in the context of its surrounding words using a particular taxonomical relation. Instead, we disambiguate a set of related words at once using a given textual context. Besides, we use a sense representation that overcomes the sparseness of WordNet relations, and that relates semantically word senses.

Our proposal relies on both topic signatures built from WordNet and the Extended Star clustering algorithm. The way this clustering algorithm relates sense representations resembles the manner in which syntactic or discourse relations link textual components.

We evaluate the proposed method according to both fine- and coarse-grained disambiguation. In the experiments carried out over Senseval-3 all-words, Semcor 2.0, and SemEval-2007 coarse-grained corpora, promising results were obtained. Our proposal achieves better recall values than other knowledge-driven disambiguation methods over the whole SemCor corpus in the disambiguation of nouns, and performs very well in the SemEval-2007 coarse-grained disambiguation task.

As further work, we plan to experiment with other levels of disambiguation such as phrases and simple sentences to explore its impact in the disambiguation task.

REFERENCES

[1] E. Agirre and O. López, "Clustering wordnet word senses," in *Proceedings of the Conference on Recent Advances on Natural Language Processing*, Bulgary, 2003, pp. 121–130.

[2] E. Agirre and G. Rigau, "Word Sense Disambiguation Using Conceptual Density," in *Proceedings of the 16th Conference on Computational Linguistic*, Vol. 1, Denmark, 1996, pp. 16–22.

[3] S. Bordag, "Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation," in *11st Conference of the European Chapter of the Association for Computational Linguistic*, Italy, 2006.

[4] D. Buscaldi and P. Rosso, "UPV-WSD: Combining different WSD Methods by means of Fuzzy Borda Voting," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, Association for Computational Linguistic, Prage, 2007, pp. 434–437.

[5] Y. Chali and S. R. Joty, "UofL: Word Sense Disambiguation Using Lexical Cohesion," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, Association for Computational Linguistic, Prage, 2007, pp. 476–479.

[6] D. Fernández-Amorós, J. Gonzalo and F. Verdejo, "The Role of Conceptual Relations in Word Sense Disambiguation," in *Proceedings of the 6th International Workshop on Applications of Natural Language for Information Systems*, Spain, 2001, pp. 87–98.

[7] R. Gil-García, J. M. Badia-Contelles and A. Pons-Porrata, "Extended Star Clustering Algorithm," *Progress in Pattern Recognition, Speech and Image Analysis*, Lecture Notes on Computer Sciences, Vol. 2905, Springer-Verlag, 2003, pp. 480-487.

[8] N. Ide and J. Veronis, "Word Sense Disambiguation: The State of the Art," *Computational Linguistics* 24:1, 1998, pp. 1–40.

[9] R. Ion and D. Tufis, "RACAI: Meaning Affinity Models," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, Association for Computational Linguistic, Prage, 2007, pp. 277–281.

[10] R. Koeling and D. McCarthy, "Sussx: WSD using Automatically Acquired Predominant Senses," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, Association for Computational Linguistic, Prage, 2007, pp. 314–317.

[11] M. Lesk, "Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone," in *Proceedings of the 5th Annual International Conference on Systems Documentation*, Canada, 1986, pp. 24–26.

[12] C.-Y. Lin and E. Hovy, "The Automated Acquisition of Topic Signatures for Text Summarization," in *Proceedings of the COLING Conference*, France, 2000, pp. 495–501.

[13] R. Mihalcea and D.I. Moldovan, "EZ. WordNet: Principles for Automatic Generation of a Coarse Grained WordNet," in *Proceedings of the FLAIRS Conference*, Florida, 2001, pp. 454–458.

[14] G. Miller, "WordNet: A Lexical Database for English," *Communications of the ACM* 38:11, 1995, pp. 39–41.

[15] A. Montoyo, A. Suárez, G. Rigau and M. Palomar, "Combining Knowledge- and Corpus-based Word-Sense-Disambiguation Methods," *Journal of Artificial Intelligence Research* 23, 2005, pp. 299–330.

[16] R. Navigli, K.C. Litkowski and O. Hargraves, "SemEval-2007 Task 07: Coarse-Grained English All-Words Task," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, Association for Computational Linguistic, Prage, 2007, pp. 30–35.

[17] C. Niu, W. Li, R. K. Srihari, H. Li and L. Crist, "Context Clustering for Word Sense Disambiguation Based on Modeling Pairwise Context Similarities," in *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Spain, 2004, pp. 187–190.

[18] T. Pedersen, A. Purandare and A. Kulkarni, "Name Discrimination by Clustering Similar Contexts," in *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, Mexico, 2005, pp. 226–237.

[19] G. Salton, A. Wong and C. S. Yang, "A Vector Space Model for Information Retrieval," *Journal of the American Society for Information Science* 18:11, 1975, pp. 613–620.

[20] B. Snyder and M. Palmer, "The English all-words task," in *Proceedings of the third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Spain, 2004, pp. 41–43.

[21] G. Udani, S. Dave, A. Davis and T. Sibley, "Noun Sense Induction Using Web Search Results," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Brazil, 2005, pp. 657–658.