

Discovering word senses from a network of lexical cooccurrences

Olivier Ferret

CEA — LIST/LIC2M

18, route du Panorama

92265 Fontenay-aux-Roses, France

ferreto@zoe.cea.fr

Abstract

Lexico-semantic networks such as WordNet have been criticized about the nature of the senses they distinguish as well as on the way they define these senses. In this article, we present a possible solution to overcome these limits by defining the sense of words from the way they are used. More precisely, we propose to differentiate the senses of a word from a network of lexical cooccurrences built from a large corpus. This method was tested both for French and English and was evaluated for English by comparing its results with WordNet.

1 Introduction

Semantic resources have proved to be useful in information retrieval and information extraction for applications such as query expansion (Voorhees, 1998), text summarization (Harabagiu and Maiorano, 2002) or question/answering (Pasca and Harabagiu, 2001). But this work has also shown that these resources must be used with caution: they bring on an improvement of results only if word sense disambiguation is performed with a great accuracy. These findings bring one of the first roles of a semantic resource to light: discriminating and characterizing the senses of a set of words. The main semantic resources with a wide coverage that can be exploited by computers are lexico-semantic networks such as WordNet. Because of the way they were built, mainly by hand, these networks are not fundamentally different from traditional dictionaries. Hence, it is not very surprising that they were criticized, as in (Harabagiu *et al.*, 1999), for not being suitable for Natural Language Processing. They were criticized both about the nature of the senses they discriminate and the way they characterize them. Their senses are considered as too fine-grained but also incomplete. Moreover, they are generally defined through their relations with synonyms, hyponyms

and hyperonyms but not by elements that describe the contexts in which they occur.

One of the solutions for solving this problem consists in automatically discovering the senses of words from corpora. Each sense is defined by a list of words that is not restricted to synonyms or hyperonyms. The work done in this area can be divided into three main trends. The first one, represented by (Pantel and Lin, 2002), is not focused on the problem of discovering word senses: its main objective is to build classes of equivalent words from a distributionalist viewpoint, hence to gather words that are mainly synonyms. In the case of (Pantel and Lin, 2002), the discovering of word senses is a side effect of the clustering algorithm, Cluster By Committee, used for building classes of words: as a word can belong to several classes, each of them can be considered as one of its senses. The second main trend, found in (Schütze, 1998), (Pedersen and Bruce, 1997) and (Purandare, 2003), represents each instance of a target word by a set of features that occur in its neighborhood and applies an unsupervised clustering algorithm to all its instances. Each cluster is then considered as a sense of the target word. The last trend, explored by (Véronis, 2003), (Dorow and Widdows, 2003) and (Rapp, 2003), starts from the cooccurents of a word recorded from a corpus and builds its senses by gathering its cooccurents according to their similarity or their dissimilarity. Our work takes place in this last trend.

2 Overview

The starting point of the method we present in this article is a network of lexical cooccurrences, that is a graph whose vertices are the significant words of a corpus and edges represent the cooccurrences between these words in the corpus. The discovering of word senses is performed word by word and the processing of a word only relies on the subgraph that contains its cooccurents. The first

step of the method consists in building a matrix of similarity between these cooccurents by exploiting their relations in the subgraph. An unsupervised clustering algorithm is then applied for grouping these cooccurents and giving rise to the senses of the considered word. This method, as the ones presented in (Véronis, 2003), (Dorow and Widdows, 2003) and (Rapp, 2003), relies on the following hypothesis: in the subgraph gathering the cooccurents of a word, the number of relations between the cooccurents defining a sense is higher than the number of relations that these cooccurents have with those defining the other senses of the considered word. The clustering algorithm that we use is an adaptation of the Shared Nearest Neighbors (SNN) algorithm presented in (Ertöz *et al.*, 2001). This algorithm particularly fits our problem as it automatically determines the number of clusters, in our case the number of senses of a word, and does not take into account the elements that are not representative of the clusters it builds. This last point is especially important for our application as there is a lot of “noise” among the cooccurents of a word.

3 Networks of lexical cooccurences

The method we present in this article for discovering word senses was applied both for French and English. Hence, two networks of lexical cooccurences were built: one for French, from the *Le Monde* newspaper (24 months between 1990 and 1994), and one for English, from the *L.A. Times* newspaper (2 years, part of the TREC corpus). The size of each corpus was around 40 million words.

The building process was the same for the two networks. First, the initial corpus was pre-processed in order to characterize texts by their topically significant words. Thus, we retained only the lemmatized form of plain words, that is, nouns, verbs and adjectives. Cooccurences were classically extracted by moving a fixed-size window on texts. Parameters were chosen in order to catch topical relations: the window was rather large, 20-word wide, and took into account the boundaries of texts; moreover, cooccurences were indifferent to word order. As (Church and Hanks, 1990), we adopted an evaluation of mutual information as a cohesion measure of each cooccurrence. This measure was normalized according to the maximal mutual information relative to the considered corpus. After filtering the less significant cooccur-

rences (cooccurences with less than 10 occurrences and cohesion lower than 0.1), we got a network with approximately 23,000 words and 5.2 million cooccurences for French, 30,000 words and 4.8 million cooccurences for English.

4 Word sense discovery algorithm

4.1 Building of the similarity matrix between cooccurents

The number and the extent of the clusters built by a clustering algorithm generally depend on a set of parameters that can be tuned in one way or another. But this possibility is implicitly limited by the similarity measure used for comparing the elements to cluster. In our case, the elements to cluster are the cooccurents in the network of lexical cooccurences of the word whose senses have to be discriminated. Within the same framework, we tested two measures for evaluating the similarity between the cooccurents of a word in order to get word senses with different levels of granularity. The first measure corresponds to the cohesion measure between words in the cooccurrence network. If there is no relation between two words in the network, the similarity is equal to zero. This measure has the advantage of being simple and efficient from an algorithmic viewpoint but some semantic relations are difficult to catch only from cooccurences in texts. For instance, we experimentally noticed that there are few synonyms of a word among its cooccurents¹. Hence, we can expect that some senses that are discriminated by the algorithm actually refer to one sense.

To overcome this difficulty, we also tested a measure that relies not only on first order cooccurences but also on second order cooccurences, which are known to be “less sparse and more robust” than first order ones (Schütze, 1998). This measure is based on the following principle: a vector whose size is equal to the number of cooccurents of the considered word is associated to each of its cooccurents. This vector contains the cohesion values between this cooccurrence and the other ones. As for the first measure, a null value is taken when there is no relation between two words in the cooccurrence network. The similarity matrix is then built by applying the cosine measure between

¹ This observation comes from the intersection, for each word of the *L.A. Times* network, of its cooccurents in the network and its synonyms in WordNet.

each couple of vectors, *i.e.* each couple of cooccurrents. With this second measure, two cooccurrents can be found strongly linked even though they are not directly linked in the cooccurrence network: they just have to share a significant number of words with which they are linked in the cooccurrence network.

4.2 The Shared Nearest Neighbors (SNN) algorithm

The SNN algorithm is representative of the algorithms that perform clustering by detecting the high-density areas of a similarity graph. In such a graph, each vertex represents an element to cluster and an edge links two vertices whose similarity is not null. In our case, the similarity graph directly corresponds to the cooccurrence network with the first order cooccurrences whereas with the second order cooccurrences, it is built from the similarity matrix described in Section 4.1. The SNN algorithm can be split up into two main steps: the first one aims at finding the elements that are the most representative of their neighborhood by masking the less important relations in the similarity graph. These elements are the seeds of the final clusters that are built in the second step by aggregating the remaining elements to those selected by the first step. More precisely, the SNN algorithm is applied to the discovering of the senses of a target word as follows:

1. sparsification of the similarity graph: for each cooccurrent of the target word, only the links towards the k ($k=15$ in our experiments) most similar other cooccurrents are kept.
2. building of the shared nearest neighbor graph: this step only consists in replacing, in the sparsified graph, the value of each edge by the number of direct neighbors shared by the two cooccurrents linked by this edge.
3. computation of the distribution of strong links among cooccurrents: as for the first step, this one is a kind of sparsification. Its aim is to help finding the seeds of the senses, *i.e.* the cooccurrents that are the most representative of a set of cooccurrents. This step is also a means for discarding the cooccurrents that have no relation with the other ones. More precisely, two cooccurrents are considered as strongly linked if the number of the neighbors they share is higher than a fixed threshold. The

higher than a fixed threshold. The number of strong links of each cooccurrent is then computed.

4. identification of the sense seeds and filtering of noise: the sense seeds and the cooccurrents to discard are determined by comparing their number of strong links with a fixed threshold.
5. building of senses: this step mainly consists in associating to the sense seeds identified by the previous step the remaining cooccurrents that are the most similar to them. The result is a set of clusters that each represents a sense of the target word. For associating a cooccurrent to a sense seed, the strength of the link between them must be higher than a given threshold. If a cooccurrent can be tied to several seeds, the one that is the most strongly linked to it is chosen. Moreover, the seeds that are considered as too close from each other for giving rise to separate senses can also be grouped during this step in accordance with the same criteria than the other cooccurrents.
6. extension of senses: after the previous steps, a set of cooccurrents that are not considered as noise are still not associated to a sense. The size of this set depends on the strictness of the threshold controlling the aggregation of a cooccurrent to a sense seed but as we are interested in getting homogeneous senses, the value of this threshold cannot be too low. Nevertheless, we are also interested in having a definition as complete as possible of each sense. As senses are defined at this point more precisely than at step 4, the integration into these senses of cooccurrents that are not strongly linked to a sense seed can be performed on a larger basis, hence in a more reliable way.

4.3 Adaptation of the SNN algorithm

For implementing the SNN algorithm presented in the previous section, one of the points that must be specified more precisely is the way its different thresholds are fixed. In our case, we chose the same method for all of them: each threshold is set as a quantile of the values it is applied to. In this way, it is adapted to the distribution of these values. For the identification of the sense seeds (threshold equal to 0.9) and for the definition of the cooccurrents that are noise (threshold equal to

	LM-1	LM-2	LAT-1	LAT-1.no	LAT-2.no
number of words	17,261	17,261	13,414	6,177	6,177
percentage of words with at least one sense	44.4%	42.7%	39.8%	41.8%	39%
average number of senses by word	2.8	2.2	1.6	1.9	1.5
average number of words describing a sense	16.1	16.3	18.7	20.2	18.9

Table 1: Statistics about the results of our word sense discovery algorithm

0.2), the thresholds are quantiles of the number of strong links of cooccurents. For defining strong links (threshold equal to 0.65), associating cooccurrents to sense seeds (threshold equal to 0.5) and aggregating cooccurrent to senses (threshold equal to 0.7), the thresholds are quantiles of the strength of the links between cooccurrents in the shared nearest neighbor graph.

We also introduced two main improvements to the SNN algorithm. The first one is the addition of a new step between the last two ones. This comes from the following observation: although a sense seed can be associated to another one during the step 5, which means that the two senses they represent are merged, some clusters that actually correspond to one sense are not merged. This problem is observed with the first and the second order cooccurrences and cannot be solved, without merging unrelated senses, only by adjusting the threshold that controls the association of a cooccurrent to a sense seed. In most of these cases, the “split” sense is scattered over one large cluster and one or several small clusters that only contain 3 or 4 cooccurrents. More precisely, the sense seeds of the small clusters are not associated to the seed of the large cluster while most of the cooccurrents that are linked to them are associated to this seed. Instead of defining a specific mechanism for dealing with these small clusters, we chose to let the SNN algorithm to solve the problem by only deleting these small clusters (size < 6) after the step 5 and marking their cooccurrents as unclassified. The last step of the algorithm aggregates in most of the cases these cooccurrents to the large cluster. Moreover, this new step makes the built senses more stable when the parameters of the algorithm are only slightly modified.

The second improvement, which has a smaller impact than the first one, aims at limiting the noise that is brought into clusters by the last step. In the algorithm of (Ertöz *et al.*, 2001), an element can be associated to a cluster when the strength of its link with one of the elements of this cluster is

higher than a given threshold. This condition is stricter in our case as it concerns the average strength of the links between the unclassified cooccurrent and those of the cluster.

5 Experiments

We applied our algorithm for discovering word senses to the two networks of lexical cooccurrences we have described in Section 3 (LM: French; LAT: English) with the parameters given in Section 4. For each network, we tested the use of first order cooccurrences (LM-1 and LAT-1) and second order ones (LM-2 and LAT-2). For English, the use of second order cooccurrences was tested only for the subpart of the words of the network that was selected for the evaluation of Section 6 (LAT-2.no). Table 1 gives some statistics about the results of the discovered senses for the different cases. We can notice that a significant percentage of words do not have any sense, even with second order cooccurrences. This comes from the fact that their cooccurrents are weakly linked to each other in the cooccurrence network they are part of, which probably means that their senses are not actually represented in this network. We can also notice that the use of second order cooccurrence actually leads to have a smaller number of senses by word, hence to have senses with a larger definition. As Véronis (2003), we give in Table 2 as an example of the results of our algorithm some of the words defining the senses of the polysemous French word *barrage*, which was part of the ROMANSEVAL evaluation. Whatever the kind of cooccurrences it relies on, our algorithm finds three of the four senses distinguished in (Véronis, 2003): *dam* (senses 1.3 and 2.1); *barricading*, *blocking* (senses 1.1, 1.2 and 2.2); *barrier*, *frontier* (senses 1.4 and 2.3). The sense *play-off game* (*match de barrage*), which refers to the domain of sport, is not found as it is weakly represented in the cooccurrence network and is linked to words, such as *division*, that are also ambiguous (it refers

LM-1	1.1	manifestant, forces_de_l'ordre, préfecture, agriculteur, protester, incendier, calme, pierre (<i>demonstrator, the police, prefecture, farmer, to protest, to burn, quietness, stone</i>)
	1.2	conducteur, routier, véhicule, poids_lourd, camion, permis, trafic, bloquer, voiture, autoroute (<i>driver, lorry driver, vehicule, lorry, truck, driving licence, traffic, to block, car, highway</i>)
	1.3	fleuve, rivière, lac, bassin, mètre_cube, crue, amont, pollution, affluent, saumon, poisson (<i>river(2), lake, basin, cubic meter, swelling, upstream water, pollution, affluent, salmon, fish</i>)
	1.4	blessé, casque_bleu, soldat, tir, milice, convoi, évacuer, croate, milicien, combattant (<i>wounded, U.N. soldier, soldier, firing, militia, convoy, to evacuate, Croatian, militiaman, combatant</i>)
LM-2	2.1	eau, mètre, lac, pluie, rivière, bassin, fleuve, site, poisson, affluent, montagne, crue, vallée (<i>water, meter, lake, rain, river(2), basin, setting, fish, affluent, mountain, swelling, valley</i>)
	2.2	conducteur, trafic, routier, route, camion, chauffeur, voiture, chauffeur_routier, poids_lourd (<i>driver, traffic, lorry driver(3), road, lorry, car, truck</i>)
	2.3	casque_bleu, soldat, tir, convoi, milicien, blindé, milice, aéroport, blessé, incident, croate (<i>U.N. soldier, soldier, firing, convoy, militiaman, tank, militia, airport, wounded, incident, Croatian</i>)

Table 2: Senses found by our algorithm for the word *barrage*

both to the sport and the military domains). It should be noted that *barrage* has only 1,104 occurrences in our corpus while it has 7,000 occurrences in the corpus of (Véronis, 2003), built by crawling from the Internet the pages found by a meta search engine queried with this word and its morphological variants. This example is also a good illustration of the difference of granularity of the senses built from first order cooccurrences and those built from the second order ones. The sense 1.1, which is close to the sense 1.2 as the two refers to demonstrations in relation to a category of workers, disappears when the second order cooccurrences are used. Table 3 gives examples of discovered senses from first order cooccurrences only, one for French (LM-1) and two for English (LAT-1).

6 Evaluation

The discovering of word senses, as most of the work dedicated to the building of linguistic resources, comes up against the problem of evaluating its results. The most direct way of doing it is to compare the resource to evaluate with a similar resource that is acknowledged as a golden standard. For word senses, the WordNet-like lexico-semantic networks can be considered as such a standard. Using this kind of networks for evaluating the word senses that we find is of course criticizable as our aim is to overcome their insufficiencies. Nevertheless, as these networks are carefully controlled, such an evaluation provides at least a first judgment about the reliability of the discovered senses. We chose to take up the evaluation method proposed in (Pantel and Lin, 2002). This method relies on WordNet and shows a rather

good agreement between its results and human judgments (88% for Pantel and Lin). As a consequence, our evaluation was done only for English, and more precisely with WordNet 1.7. For each considered word, the evaluation method tries to map one of its discovered senses with one of its synsets in WordNet by applying a specific similarity measure. Hence, only the precision of the word sense discovering algorithm is evaluated but Pantel and Lin indicate that recall is not very significant in this context: a discovered sense may be correct and not present in WordNet and conversely, some senses in WordNet are very close and should be joined for most of the applications using WordNet. They define a recall measure but only for ranking the results of a set of systems. Hence, it cannot be applied in our case.

The similarity measure between a sense and a synset used for computing precision relies on the Lin's similarity measure between two synsets:

$$sim(s1, s2) = \frac{2 \times \log P(s)}{\log P(s1) + \log P(s2)} \quad (1)$$

where s is the most specific synset that subsumes $s1$ and $s2$ in the WordNet hierarchy and $P(s)$ represents the probability of the synset s estimated from a reference corpus, in this case the SemCor corpus. We used the implementation of this measure provided by the Perl module WordNet::Similarity v0.06 (Patwardhan and Pedersen, 2003). The similarity between a sense and a synset is more precisely defined as the average value of the similarity values between the words that characterize the sense, or a subset of them, and the synset. The similarity between a word and a synset

organe (1300)²	patient, transplantation, greffe, malade, thérapeutique, médical, médecine, greffer, rein (<i>patient, transplantation, transplant, sick person, therapeutic, medical, medicine, to transplant, kidney</i>) procréation, embryon, éthique, humain, relatif, bioéthique, corps_humain, gène, cellule (<i>procreation, embryo, ethical, human, relative, bioethics, human body, gene, cell</i>) constitutionnel, consultatif, constitution, instituer, exécutif, législatif, siéger, disposition (<i>constitutional, consultative, constitution, to institute, executive, legislative, to sit, clause</i>) article, hebdomadaire, publication, rédaction, quotidien, journal, éditorial, rédacteur (<i>article, weekly, publication, editorial staff, daily, newspaper, editorial, sub-editor</i>)
mouse (563)	compatible, software, computer, machine, user, desktop, pc, graphics, keyboard, device laboratory, researcher, cell, gene, generic, human, hormone, research, scientist, rat
party (16999)	candidate, démocrate, républicain, gubernatorial, présidentielle, partisan, réajustement ballroom, cocktail, champagne, guest, bash, gala, wedding, birthday, invitation, festivity caterer, uninvited, party-goers, black-tie, hostess, buffet, glitches, napkins, catering

Table 3: Senses found by our algorithm from first order cooccurrences (LM-1 and LAT-1)

is equal to the highest similarity value among those between the synset and the synsets to which the word belongs to. Each of these values is given by (1). A sense is mapped to the synset that is the most similar to it, providing that the similarity between them is higher than a fixed threshold (equal to 0.25 as in (Pantel and Lin, 2002)). Finally, the precision for a word is given by the proportion of its senses that match one of its synsets.

Table 4 gives the results of the evaluation of our algorithm for the words of the English cooccurrence network that are nouns only and for which at least one sense was discovered. As Pantel and Lin, we only take into account for evaluation 4 words of each sense, whatever the number of words that define it. But, because of the way our senses are built, we have not a specific measure of the similarity between a word and the words that characterize its senses. Hence, we computed two variants of the precision measure. The first one selects the four words of each sense by relying on their number of strong links in the shared nearest neighbor graph. The second one selects the four words that have the highest similarity score with one of the synsets of the target word, which is called “optimal choice” in Table 4³. A clear difference can be noted between the two variants. With the optimal choice of the four words, we get results that are similar to those of Pantel and Lin: their precision is equal to 60.8 with an average number of words defining a sense equal to 14.

On the other hand, Table 4 shows that the words selected on the basis of their number of strong links are not strongly linked in WordNet (according to Lin’s measure) to their target word. This does not mean that the selected words are not interesting for describing the senses of the target word but more probably that the semantic relations that they share with the target word are different from hyperonymy. The results of Pantel and Lin can be explained by the fact that their algorithm is based on the clustering of similar words, *i.e.* words that are likely to be synonyms, and not on the clustering of the cooccurrences of a word, which are not often synonyms of that word. Moreover, their initial corpus is much larger (around 144 millions words) than ours and they make use of more elaborated tools, such as a syntactic analyzer.

	LAT-1.no	LAT-2.no
number of strong links	19.4	20.8
optimal choice	56.2	63.7

Table 4: Average precision of discovered senses for English in relation with WordNet

As expected, the results obtained with first order cooccurrences (LAT-1.no), which produce a higher number of senses by word, are lower than the results obtained with second order cooccurrences (LAT-2.no). However, without a recall measure, it is difficult to draw a clear conclusion from this observation: some senses of LAT-1.no probably result from the artificial division of an actual word sense but the fact to have more homogeneous senses in LAT-2.no also facilitates in this case the mapping with WordNet’s synsets.

² Each word is given with its frequency in the corpus used for building the cooccurrence network.

³ This selection procedure is only used for evaluation and we do not rely on WordNet for building our senses.

7 Related work

As they rely on the detection of high-density areas in a network of cooccurrences, (Véronis, 2003) and (Dorow and Widdows, 2003) are the closest methods to ours. Nevertheless, two main differences can be noted with our work. The first one concerns the direct use they make of the network of cooccurrences. In our case, we chose a more general approach by working at the level of a similarity graph: when the similarity of two words is given by their relation of cooccurrence, our situation is comparable to the one of (Véronis, 2003) and (Dorow and Widdows, 2003); but in the same framework, we can also take into account other kinds of similarity relations, such as the second order cooccurrences.

The second main difference is the fact they discriminate senses in an iterative way. This approach consists in selecting at each step the most obvious sense and then, to update the graph of cooccurrences by discarding the words that make up the new sense. The other senses are then easier to discriminate. We preferred to put emphasis on the ability to gather close or identical senses that are artificially distinguished (see Section 4.3). From a global viewpoint, these two differences lead (Véronis, 2003) and (Dorow and Widdows, 2003) to build finer senses than ours. Nevertheless, as methods for discovering word senses from a corpus tend to find a too large number of close senses, it was more important from our viewpoint to favour the building of stable senses with a clear definition rather than to discriminate very fine senses.

8 Conclusion and future work

In this article, we have presented a new method for discriminating and defining the senses of a word from a network of lexical cooccurrences. This method consists in applying an unsupervised clustering algorithm, in this case the SNN algorithm, to the cooccurrences of the word by relying on the relations that these cooccurrences have in the cooccurrence network. We have achieved a first evaluation based on the methodology defined in (Pantel and Lin, 2002). This evaluation has shown that in comparison with WordNet taken as a reference, the relevance of the discriminated senses is comparable to the relevance of Pantel and Lin's word senses. But it has also shown that the similarity between a discovered sense and a synset

ilarity between a discovered sense and a synset of WordNet must be evaluated in our case by taking into account a larger set of semantic relations, especially those implicitly present in the glosses. Moreover, an evaluation based on the use of the built senses in an application such as query expansion is necessary to determine the actual interest of this kind of resources in comparison with a lexico-semantic network such as WordNet.

References

- K.W. Church and P. Hanks. 1990. Word Association Norms, Mutual Information, And Lexicography. *Computational Linguistics*, 16(1): 22–29.
- Dorow B. and D. Widdows. 2003. Discovering Corpus-Specific Word Senses. In *EACL 2003*.
- L. Ertöz, M. Steinbach and V. Kumar. 2001. Finding Topics in Collections of Documents: A Shared Nearest Neighbor Approach. In *Text Mine'01, Workshop of the 1st SIAM International Conference on Data Mining*.
- S. Harabagiu and S. Maiorano. 2002. Multi-Document Summarization with GISTEXTER. In *LREC 2002*.
- S Harabagiu, G.A. Miller and D. Moldovan. 1999. WordNet 2 - A Morphologically and Semantically Enhanced Resource. In *SIGLEX'99*.
- M. Pasca and S. Harabagiu. 2001. The informative role of WordNet in Open-Domain Question Answering. In *NAACL 2001 Workshop on WordNet and Other Lexical Resources*.
- P. Pantel and D. Lin. 2002. Discovering Word Senses from Text. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2002*.
- S. Patwardhan and T. Pedersen. 2003. WordNet::Similarity, <http://www.d.umn.edu/~tpederse/similarity.html>.
- T. Pedersen and R. Bruce. 1997. Distinguishing Word Senses in Untagged Text. In *EMNLP'97*.
- A. Purandare. 2003. Discriminating Among Word Senses Using Mcquitty's Similarity Analysis. In *HLT-NAACL 03 - Student Research Workshop*.
- R. Rapp. 2003. Word Sense Discovery Based on Sense Descriptor Dissimilarity. In *Machine Translation Summit IX*.
- H. Schütze. 1998. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1): 97-123.
- J. Véronis. 2003. Cartographie lexicale pour la recherche d'information. In *TALN 2003*.
- E.M. Voorhees. 1998. *Using WordNet for text retrieval*, In "WordNet: An Electronic Lexical Database", Cambridge, MA, MIT Press, pages 285-303.