

Combining Knowledge-based Methods and Supervised Learning for Effective Italian Word Sense Disambiguation

Pierpaolo Basile
Marco de Gemmis
Pasquale Lops
Giovanni Semeraro
University of Bari (Italy)
email: basilepp@di.uniba.it

Abstract

This paper presents a WSD strategy which combines a knowledge-based method that exploits sense definitions in a dictionary and relations among senses in a semantic network, with supervised learning methods on annotated corpora. The idea behind the approach is that the knowledge-based method can cope with the possible lack of training data, while supervised learning can improve the precision of a knowledge-based method when training data are available. This makes the proposed method suitable for disambiguation of languages for which the available resources are lacking in training data or sense definitions. In order to evaluate the effectiveness of the proposed approach, experimental sessions were carried out on the dataset used for the WSD task in the EVALITA 2007 initiative, devoted to the evaluation of Natural Language Processing tools for Italian. The most effective hybrid WSD strategy is the one that integrates the knowledge-based approach into the supervised learning method, which outperforms both methods taken singularly.

1 Background and Motivations

The inherent ambiguity of human language is a greatly debated problem in many research areas, such as information retrieval and text categorization, since the presence of polysemous words might result in a wrong relevance judgment or classification of documents. These problems call for alternative methods that work not only at the lexical level of the documents, but also at the *meaning* level.

The task of Word Sense Disambiguation (WSD) consists in assigning the most appropriate meaning to a polysemous word within a given context. Applications such as machine translation, knowledge acquisition, common sense reasoning and others, require knowledge about word meanings, and WSD is essential for all these applications. The assignment of senses to words is accomplished by using two major sources of information (Nancy and Véronis, 1998):

1. the *context* of the word to be disambiguated, e.g. information contained within the text in which the word appears;
2. *external knowledge sources*, including lexical resources, as well as hand-devised knowledge sources, which provide data useful to associate words with senses.

All disambiguation work involves matching the context of the instance of the word to be disambiguated with either information from an external knowledge source (also known as *knowledge-driven WSD*), or information about the contexts of previously disambiguated instances of the word derived from corpora (*data-driven* or *corpus-based WSD*).

Corpus-based WSD exploits semantically annotated corpora to train machine learning algorithms to decide which word sense to choose in which context. Words in such annotated corpora are tagged manually using semantic classes chosen from a particular lexical semantic resource (e.g. WORDNET (Fellbaum, 1998)). Each sense-tagged occurrence of a particular word is transformed into a feature vector, which is then used in an automatic learning process. The applicability of such supervised algorithms is limited to those few words for which sense tagged data are available, and their accuracy is strongly influenced by the amount of labeled data available.

Knowledge-based WSD has the advantage of avoiding the need of sense-annotated data, rather it exploits lexical knowledge stored in machine-readable dictionaries or thesauri. Systems adopting this approach have proved to be ready-to-use and scalable, but in general they reach lower precision than corpus-based WSD systems.

Our hypothesis is that the combination of both types of strategies can improve WSD effectiveness, because knowledge-based methods can cope with the possible lack of training data, while supervised learning can improve the precision of knowledge-based methods when training data are available.

This paper presents a method for solving the semantic ambiguity of *all words* contained in a text¹. We propose a hybrid WSD algorithm that combines a knowledge-based WSD algorithm, called JIGSAW, which we designed to work by exploiting WORDNET-like dictionaries as sense repository, with a supervised machine learning

¹*all words* task tries to disambiguate all the words in a text, while *lexical sample* task tries to disambiguate only specific words

algorithm (K-Nearest Neighbor classifier). WORDNET-like dictionaries are used because they combine the characteristics of both a dictionary and a structured semantic network, supplying definitions for the different senses of words and defining groups of synonymous words by means of *synsets*, which represent distinct lexical concepts. WORDNET also organizes synsets in a conceptual structure by defining a number of semantic relationships (IS-A, PART-OF, etc.) among them.

Mainly, the paper concentrates on two investigations:

1. First, corpus-based WSD is applied to words for which training examples are provided, then JIGSAW is applied to words not covered in the first step, with the advantage of knowing the senses of the context words already disambiguated in the first step;
2. First, JIGSAW is applied to assign the most appropriate sense to those words that can be disambiguated with a high level of confidence (by setting a specific parameter in the algorithm), then the remaining words are disambiguated by the corpus-based method.

The paper is organized as follows: After a brief discussion about the main works related to our research, Section 3 gives the main ideas underlying the proposed hybrid WSD strategy. More details about the K-NN classification algorithm and JIGSAW, on which the hybrid WSD approach is based, are provided in Section 4 and Section 5, respectively. Experimental sessions have been carried out in order to evaluate the proposed approach in the critical situation when training data are not much reliable, as for Italian. Results are presented in Section 6, while conclusions and future work close the paper.

2 Related Work

For some Natural Language Processing (NLP) tasks, such as part of speech tagging or named entity recognition, there is a consensus on what makes a successful algorithm, regardless of the approach considered. Instead, no such consensus has been reached yet for the task of WSD, and previous work has considered a range of knowledge sources, such as local collocational clues, common membership in semantically or topically related word classes, semantic density, and others. In recent SENSEVAL-3 evaluations², the most successful approaches for *all words* WSD relied on information drawn from annotated corpora. The system developed by Decadt et al. (2002) uses two cascaded memory-based classifiers, combined with the use of a genetic algorithm for joint parameter optimization and feature selection. A separate word expert is learned for each ambiguous word, using a concatenated corpus of English sense tagged texts, including SemCor, SENSEVAL datasets, and a corpus built from WORDNET examples. The performance of this system on the SENSEVAL-3 English all words dataset was evaluated at 65.2%. Another top ranked system is the one developed by Yuret (2004), which combines two Naïve Bayes statistical models, one based on surrounding collocations and another one based on a bag of words around the target word. The statistical models are built based on SemCor and WORDNET, for an overall disambiguation accuracy of 64.1%. All previous systems use supervised methods, thus

²<http://www.senseval.org>

requiring a large amount of human intervention to annotate the training data. In the context of the current multilingual society, this strong requirement is even increased, since the so-called “sense-tagged data bottleneck problem” is emphasized.

To address this problem, different methods have been proposed. This includes the automatic generation of sense-tagged data using monosemous relatives (Leacock et al., 1998), automatically bootstrapped disambiguation patterns (Mihalcea, 2002), parallel texts as a way to point out word senses bearing different translations in a second language (Diab, 2004), and the use of volunteer contributions over the Web (Mihalcea and Chklovski, 2003). More recently, Wikipedia has been used as a source of sense annotations for building a sense annotated corpus which can be used to train accurate sense classifiers (Mihalcea, 2007). Even though the Wikipedia-based sense annotations were found reliable, leading to accurate sense classifiers, one of the limitations of the approach is that definitions and annotations in Wikipedia are available almost exclusively for nouns.

On the other hand, the increasing availability of large-scale rich (lexical) knowledge resources seems to provide new challenges to knowledge-based approaches (Navigli and Velardi, 2005; Mihalcea, 2005). Our hypothesis is that the complementarity of knowledge-based methods and corpus-based ones is the key to improve WSD effectiveness. The aim of the paper is to define a cascade hybrid method able to exploit both linguistic information coming from WORDNET-like dictionaries and statistical information coming from sense-annotated corpora.

3 A Hybrid Strategy for WSD

The goal of WSD algorithms consists in assigning a word w_i occurring in a document d with its appropriate meaning or sense s . The sense s is selected from a predefined set of possibilities, usually known as *sense inventory*. We adopt ITALWORDNET (Roventini et al., 2003) as sense repository. The algorithm is composed by two procedures:

1. **JIGSAW** - It is a knowledge-based WSD algorithm based on the assumption that the adoption of different strategies depending on Part-of-Speech (PoS) is better than using always the same strategy. A brief description of JIGSAW is given in Section 5, more details are reported in Basile et al. (2007b), Basile et al. (2007a) and Semeraro et al. (2007).
2. **Supervised learning procedure** - A K-NN classifier (Mitchell, 1997), trained on MultiSemCor corpus³ is adopted. Details are given in Section 4. MultiSemCor is an English/Italian parallel corpus, aligned at the word level and annotated with PoS, lemma and word senses. The parallel corpus is created by exploiting the SemCor corpus⁴, which is a subset of the English Brown corpus containing about 700,000 running words. In SemCor, all the words are tagged by PoS, and more than 200,000 content words are also lemmatized and sense-tagged with reference to the WORDNET lexical database. SemCor has been used in several supervised WSD algorithms for English with good results. MultiSemCor contains less annotations than SemCor, thus the accuracy and the coverage of the supervised learning for Italian might be affected by poor training data.

³<http://multisemcor.itc.it/>

⁴<http://www.cs.unt.edu/~rada/downloads.html#semcor>

The idea is to combine both procedures in a hybrid WSD approach. A first choice might be the adoption of the supervised method as first attempt, then JIGSAW could be applied to words not covered in the first step. Differently, JIGSAW might be applied first, then leaving the supervised approach to disambiguate the remaining words. An investigation is required in order to choose the most effective combination.

4 Supervised Learning Method

The goal of supervised methods is to use a set of annotated data as little as possible, and at the same time to make the algorithm general enough to be able to disambiguate all content words in a text. We use MultiSemCor as annotated corpus, since at present it is the only available semantic annotated resource for Italian. The algorithm starts with a preprocessing stage, where the text is tokenized, stemmed, lemmatized and annotated with PoS.

Also, the collocations are identified using a sliding window approach, where a collocation is considered to be a sequence of words that forms a compound concept defined in ITALWORDNET (e.g. artificial intelligence). In the training step, a semantic model is learned for each PoS, starting with the annotated corpus. These models are then used to disambiguate words in the test corpus by annotating them with their corresponding meaning. The models can only handle words that were previously seen in the training corpus, and therefore their coverage is not 100%. Starting with an annotated corpus formed by all annotated files in MultiSemCor, a separate training dataset is built for each PoS. For each open-class word in the training corpus, a feature vector is built and added to the corresponding training set. The following features are used to describe an occurrence of a word in the training corpus as in Hoste et al. (2002):

- **Nouns** - 2 features are included in feature vector: the first noun, verb, or adjective before the target noun, within a window of at most three words to the left, and its PoS;
- **Verbs** - 4 features are included in feature vector: the first word before and the first word after the target verb, and their PoS;
- **Adjectives** - all the nouns occurring in two windows, each one of six words (before and after the target adjective) are included in the feature vector;
- **Adverbs** - the same as for adjectives, but vectors contain adjectives rather than nouns.

The label of each feature vector consists of the target word and the corresponding sense, represented as *word#sense*. Table 1 describes the number of vectors for each PoS.

To annotate (disambiguate) new text, similar vectors are built for all content-words in the text to be analyzed. Consider the target word *bank*, used as a noun. The algorithm catches all the feature vectors of *bank* as a noun from the training model, and builds the feature vector v_f for the target word. Then, the algorithm computes the similarity between each training vector and v_f and ranks the training vectors in decreasing order according to the similarity value.

Table 1: Number of feature vectors

PoS	#feature vectors
Noun	38,546
Verb	18,688
Adjective	6,253
Adverb	1,576

The similarity is computed as Euclidean distance between vectors, where POS distance is set to 1, if POS tags are different, otherwise it is set to 0. Word distances are computed by using the *Levenshtein* metric, that measures the amount of difference between two strings as the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character (Levenshtein, 1966). Finally, the target word is labeled with the most frequent sense in the first K vectors.

5 JIGSAW - Knowledge-based Approach

JIGSAW is a WSD algorithm based on the idea of combining three different strategies to disambiguate nouns, verbs, adjectives and adverbs. The main motivation behind our approach is that the effectiveness of a WSD algorithm is strongly influenced by the POS tag of the target word.

JIGSAW takes as input a document $d = (w_1, w_2, \dots, w_h)$ and returns a list of synsets $X = (s_1, s_2, \dots, s_k)$ in which each element s_i is obtained by disambiguating the *target word* w_i based on the information obtained from the sense repository about a few immediately surrounding words. We define the *context* C of the target word to be a window of n words to the left and another n words to the right, for a total of $2n$ surrounding words. The algorithm is based on three different procedures for nouns, verbs, adverbs and adjectives, called $JIGSAW_{nouns}$, $JIGSAW_{verbs}$, $JIGSAW_{others}$, respectively.

JIGSAW_{nouns} - Given a set of nouns $W = \{w_1, w_2, \dots, w_n\}$, obtained from document d , with each w_i having an associated sense inventory $S_i = \{s_{i1}, s_{i2}, \dots, s_{ik}\}$ of possible senses, the goal is assigning each w_i with the most appropriate sense $s_{ih} \in S_i$, according to the *similarity* of w_i with the other words in W (the context for w_i). The idea is to define a function $\phi(w_i, s_{ij})$, $w_i \in W$, $s_{ij} \in S_i$, that computes a value in $[0, 1]$ representing the confidence with which word w_i can be assigned with sense s_{ij} . In order to measure the relatedness of two words we adopted a modified version of the Leacock and Chodorow (1998) measure, which computes the length of the path between two concepts in a hierarchy by passing through their *Most Specific Subsumer* (MSS). We introduced a constant factor *depth* which limits the search for the MSS to *depth* ancestors, in order to avoid “poorly informative” MSSs. Moreover, in the similarity computation, we introduced both a Gaussian factor $G(pos(w_i), pos(w_j))$, which takes into account the distance between the position of the words in the text to be disambiguated, and a factor $R(k)$, which assigns s_{ik} with a numerical value, according to the frequency score in ITALWORDNET.

JIGSAW_{verbs} - We define the *description* of a synset as the string obtained by

concatenating the gloss and the sentences that ITALWORDNET uses to explain the usage of a synset. $JIGSAW_{verbs}$ includes, in the context C for the target verb w_i , all the nouns in the window of $2n$ words surrounding w_i . For each candidate synset s_{ik} of w_i , the algorithm computes $nouns(i, k)$, that is the set of nouns in the description for s_{ik} . Then, for each w_j in C and each synset s_{ik} , the following value is computed:

$$(1) \quad \max_{jk} = \max_{w_l \in nouns(i, k)} \{ \text{sim}(w_j, w_l, \text{depth}) \}$$

where $\text{sim}(w_j, w_l, \text{depth})$ is the same similarity measure adopted by $JIGSAW_{nouns}$. Finally, an overall similarity score among s_{ik} and the whole context C is computed:

$$(2) \quad \varphi(i, k) = R(k) \cdot \frac{\sum_{w_j \in C} G(\text{pos}(w_i), \text{pos}(w_j)) \cdot \max_{jk}}{\sum_h G(\text{pos}(w_i), \text{pos}(w_h))}$$

where both $R(k)$ and $G(\text{pos}(w_i), \text{pos}(w_j))$, that gives a higher weight to words closer to the target word, are defined as in $JIGSAW_{nouns}$. The synset assigned to w_i is the one with the highest φ value.

JIGSAW_{others} - This procedure is based on the WSD algorithm proposed in Banerjee and Pedersen (2002). The idea is to compare the glosses of each candidate sense for the target word to the glosses of all the words in its context.

6 Experiments

The main goal of our investigation is to study the behavior of the hybrid algorithm when available training resources are not much reliable, e.g. when a lower number of sense descriptions is available, as for Italian. The hypothesis we want to evaluate is that corpus-based methods and knowledge-based ones can be combined to improve the accuracy of each single strategy.

Experiments have been performed on a standard test collection in the context of the *All-Words-Task*, in which WSD algorithms attempt to disambiguate all words in a text. Specifically, we used the EVALITA WSD All-Words-Task dataset⁵, which consists of about 5,000 words labeled with ITALWORDNET synsets.

An important concern for the evaluation of WSD systems is the agreement rate between human annotators on word sense assignment.

While for natural language subtasks like part-of-speech tagging, there are relatively well defined and agreed-upon criteria of what it means to have the “correct” part of speech assigned to a word, this is not the case for word sense assignment. Two human annotators may genuinely disagree on their sense assignment to a word in a context, since the distinction between the different senses for a commonly used word in a dictionary like WORDNET tend to be rather fine.

What we would like to underline here is that it is important that human agreement on an annotated corpus is carefully measured, in order to set an upper bound to the performance measures: it would be futile to expect computers to agree more with the reference corpus than human annotators among them. For example, the inter-annotator agreement rate during the preparation of the SENSEVAL-3 WSD English All-Words-Task dataset (Agirre et al., 2007) was approximately 72.5%.

⁵<http://evalita.itc.it/tasks/wsd.html>

Unfortunately, for EVALITA dataset, the inter-annotator agreement has not been measured, one of the reasons why the evaluation for Italian WSD is very hard. In our experiments, we reasonably selected different baselines to compare the performance of the proposed hybrid algorithm.

6.1 Integrating JIGSAW into a supervised learning method

The design of the experiment is as follows: firstly, corpus-based WSD is applied to words for which training examples are provided, then JIGSAW is applied to words not covered by the first step, with the advantage of knowing the senses of the context words already disambiguated in the first step. The performance of the hybrid method was measured in terms of precision (P), recall (R), F-measure (F) and the percentage A of disambiguation attempts, computed by counting the words for which a disambiguation attempt is made (the words with no training examples or sense definitions cannot be disambiguated). Table 2 shows the baselines chosen to compare the hybrid WSD algorithm on the All-Words-Task experiments.

Table 2: Baselines for Italian All-Words-Task

Setting	P	R	F	A
1^{st} sense	58.45	48.58	53.06	83.11
Random	43.55	35.88	39.34	83.11
JIGSAW	55.14	45.83	50.05	83.11
K-NN	59.15	11.46	19.20	19.38
K-NN + 1^{st} sense	57.53	47.81	52.22	83.11

The simplest baseline consists in assigning a random sense to each word (*Random*), another common baseline in Word Sense Disambiguation is first sense (1^{st} sense): each word is tagged using the first sense in ITALWORDNET that is the most commonly (frequent) used sense. The other baselines are the two methods combined in the hybrid WSD, taken separately, namely JIGSAW and K-NN, and the basic hybrid algorithm “K-NN + 1^{st} sense”, which applies the supervised method, and then adopts the first sense heuristic for the words without examples into training data. The K-NN baseline achieves the highest precision, but the lowest recall due to the low coverage in the training data (19.38%) makes this method useless for all practical purposes. Notice that JIGSAW was the only participant to EVALITA WSD All-Words-Task, therefore it currently represents the only available system performing WSD All-Words task for the Italian language.

Table 3: Experimental results of K-NN+JIGSAW

Setting	P	R	F	A
K-NN + JIGSAW	56.62	47.05	51.39	83.11
K-NN + JIGSAW ($\varphi \geq 0.90$)	61.88	26.16	36.77	42.60
K-NN + JIGSAW ($\varphi \geq 0.80$)	61.40	32.21	42.25	52.06
K-NN + JIGSAW ($\varphi \geq 0.70$)	60.02	36.29	45.23	60.46
K-NN + JIGSAW ($\varphi \geq 0.50$)	59.58	37.38	45.93	62.74

Table 3 reports the results obtained by the hybrid method on the EVALITA dataset. We study the behavior of the hybrid approach with relation to that of JIGSAW, since this specific experiment aims at evaluating the potential improvements due to the inclusion of JIGSAW into K-NN. Different runs of the hybrid method have been performed, each run corresponding to setting a specific value for ϕ (the confidence with which a word w_i is correctly disambiguated by JIGSAW). In each different run, the disambiguation carried out by JIGSAW is considered reliable only when ϕ values exceed a certain threshold, otherwise any sense is assigned to the target word (this the reason why A decreases by setting higher values for ϕ).

A positive effect on precision can be noticed by varying ϕ between 0.50 and 0.90. It tends to grow and overcomes all the baselines, but a corresponding decrease of recall is observed, as a consequence of more severe constraints set on ϕ . Anyway, recall is still too low to be acceptable.

Better results are achieved when no restriction is set on ϕ (K-NN+JIGSAW in Table 3): the recall is significantly higher than that obtained in the other runs. On the other hand, the precision reached in this run is lower than in the others, but it is still acceptable.

To sum up, two main conclusions can be drawn from the experiments:

- when no constraint is set on the knowledge-based method, the hybrid algorithm K-NN+JIGSAW in general outperforms both JIGSAW and K-NN taken singularly (F values highlighted in bold in Tables 3 and 4);
- when thresholding is introduced on ϕ , no improvement is observed on the whole compared to K-NN+JIGSAW.

A deep analysis of results revealed that lower recall was achieved for verbs and adjectives rather than for nouns. Indeed, disambiguation of Italian verbs and adjectives is very hard, but the lower recall is probability due also to the fact that *JIGSAW* uses glosses for verbs and adjectives disambiguation. As a consequence, the performance depends on the accuracy of word descriptions in the glosses, while for nouns the algorithm relies only the semantic relations between synsets.

6.2 Integrating supervised learning into JIGSAW

In this experiment we test whether the supervised algorithm can help JIGSAW to disambiguate more accurately. The experiment has been organized as follows: JIGSAW is applied to assign the most appropriate sense to the words which can be disambiguated with a high level of confidence (by setting the ϕ threshold), then the remaining words are disambiguated by the K-NN classifier. The dataset and the baselines are the same as in Section 6.1.

Note that, differently from the experiments described in Table 3, run JIGSAW+K-NN has not been reported since JIGSAW covered all the target words in the first step of the cascade hybrid method, then the K-NN method is not applied at all. Therefore, for this run, results obtained by JIGSAW+K-NN correspond to those get by JIGSAW alone (reported in Table 2).

Table 4 reports the results of all the runs. Results are very similar to those obtained in the runs K-NN+JIGSAW with the same settings on ϕ . Precision tends to grow,

Table 4: Experimental results of JIGSAW+K-NN

Setting	P	R	F	A
JIGSAW ($\phi \geq 0.90$) + K-NN	61.48	27.42	37.92	44.61
JIGSAW ($\phi \geq 0.80$) + K-NN	61.17	32.59	42.52	53.28
JIGSAW ($\phi \geq 0.70$) + K-NN	59.44	36.56	45.27	61.52

while a corresponding decrease in recall is observed. The main outcome is that the overall accuracy of the best combination JIGSAW+K-NN ($\phi \geq 0.70$, F value highlighted in bold in Table 4) is outperformed by K-NN+JIGSAW. Indeed, this result was largely expected because the small size of the training set does not allow to cover words not disambiguated by JIGSAW.

Even if K-NN+JIGSAW is not able to achieve the baselines set on the 1st sense heuristic (first and last row in Table 2), we can conclude that a step toward these hard baselines has been moved. The main outcome of the study is that the best hybrid method on which further investigations are possible is K-NN+JIGSAW.

7 Conclusions and Future Work

This paper presented a method for solving the semantic ambiguity of *all words* contained in a text. We proposed a hybrid WSD algorithm that combines a knowledge-based WSD algorithm, called JIGSAW, which we designed to work by exploiting WORDNET-like dictionaries as sense repository, with a supervised machine learning algorithm (K-Nearest Neighbor classifier). The idea behind the proposed approach is that JIGSAW can cope with the possible lack of training data, while K-NN can improve the precision of JIGSAW method when training data are available. This makes the proposed method suitable for disambiguation of languages for which the available resources are lacking in training data or sense definitions, such as Italian.

Extensive experimental sessions were performed on the EVALITA WSD All-Words-Task dataset, the only dataset available for the evaluation of WSD systems for the Italian language. An investigation was carried out in order to evaluate several combinations of JIGSAW and K-NN. The main outcome is that the most effective hybrid WSD strategy is the one that runs JIGSAW after K-NN, which outperforms both JIGSAW and K-NN taken singularly. Future work includes new experiments with other combination methods, for example the *JIGSAW* output could be used as feature into supervised system or other different supervised methods could be exploited.

References

- Agirre, E., B. Magnini, O. L. de Lacalle, A. Otegi, G. Rigau, and P. Vossen (2007). SemEval-2007 Task 1: Evaluating WSD on Cross-Language Information Retrieval. In *Proceedings of SemEval-2007*. Association for Computational Linguistics.
- Banerjee, S. and T. Pedersen (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *CICLing '02: Proceedings of the Third International*

- Conference on Computational Linguistics and Intelligent Text Processing*, London, UK, pp. 136–145. Springer-Verlag.
- Basile, P., M. de Gemmis, A. Gentile, P. Lops, and G. Semeraro (2007a). JIGSAW algorithm for Word Sense Disambiguation. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*, pp. 398–401. ACL press.
- Basile, P., M. de Gemmis, A. L. Gentile, P. Lops, and G. Semeraro (2007b). The JIGSAW Algorithm for Word Sense Disambiguation and Semantic Indexing of Documents. In R. Basili and M. T. Paziienza (Eds.), *AI*IA*, Volume 4733 of *Lecture Notes in Computer Science*, pp. 314–325. Springer.
- Decadt, B., V. Hoste, W. Daelemans, and A. V. den Bosch (2002). Gambl, Genetic Algorithm optimization of Memory-based WSD. In *Senseval-3: 3th International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.
- Diab, M. (2004). Relieving the data acquisition bottleneck in word sense disambiguation. In *Proceedings of ACL*. Barcelona, Spain.
- Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.
- Hoste, V., W. Daelemans, I. Hendrickx, and A. van den Bosch (2002). Evaluating the results of a memory-based word-expert approach to unrestricted word sense disambiguation. In *Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions*, Volume 8, pp. 95–101. Association for Computational Linguistics Morristown, NJ, USA.
- Leacock, C. and M. Chodorow (1998). *Combining local context and WordNet similarity for word sense identification*, pp. 305–332. MIT Press.
- Leacock, C., M. Chodorow, and G. Miller (1998). Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics* 24(1), 147–165.
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10(8), 707–710.
- Mihalcea, R. (2002). Bootstrapping large sense tagged corpora. In *Proceedings of the 3rd International Conference on Language Resources and Evaluations*.
- Mihalcea, R. (2005). Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Morristown, NJ, USA, pp. 411–418. Association for Computational Linguistics.
- Mihalcea, R. (2007). Using Wikipedia for Automatic Word Sense Disambiguation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.
- Mihalcea, R. and T. Chklovski (2003). Open Mind Word Expert: Creating Large Annotated Data Collections with Web Users' Help. In *Proceedings of the EACL Workshop on Linguistically Annotated Corpora, Budapest*.

- Mitchell, T. (1997). *Machine Learning*. New York: McGraw-Hill.
- Nancy, I. and J. Véronis (1998). Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics* 24(1), 1–40.
- Navigli, R. and P. Velardi (2005). Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(7), 1075–1086.
- Roventini, A., A. Alonge, F. Bertagna, N. Calzolari, J. Cancila, C. Girardi, B. Magnini, R. Marinelli, M. Speranza, and A. Zampolli (2003). ItalWordNet: building a large semantic database for the automatic treatment of Italian. *Computational Linguistics in Pisa - Linguistica Computazionale a Pisa. Linguistica Computazionale, Special Issue XVIII-XIX, Tomo II*, 745–791.
- Semeraro, G., M. Deggemmis, P. Lops, and P. Basile (2007). Combining learning and word sense disambiguation for intelligent user profiling. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence IJCAI-07*, pp. 2856–2861. M. Kaufmann, San Francisco, California. ISBN: 978-1-57735-298-3.
- Yuret, D. (2004). Some experiments with a naive bayes WSD system. In *Senseval-3: 3th Internat. Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.