Optimizing Classifier Performance in Word Sense Disambiguation by Redefining Word Sense Classes

Upali S. Kohomban **Wee Sun Lee**
Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
{upali,leews}@comp.nus.edu.sg

Abstract

Learning word sense classes has been shown to be useful in fine-grained word sense disambiguation [Kohomban and Lee, 2005]. However, the common choice for sense classes, WordNet lexicographer files, are not designed for machine learning based word sense disambiguation. In this work, we explore the use of clustering techniques in an effort to construct sense classes that are more suitable for word sense disambiguation end-task. Our results show that these classes can significantly improve classifier performance over the state of the art results of unrestricted word sense disambiguation.

1 Introduction

Perhaps the most serious problem faced by research in Word Sense Disambiguation (**WSD**) is acquiring labeled training data for supervised learning. This is a crucial problem, since no system with unsupervised learning has shown comparable results to those of supervised systems, and labeling data for WSD is labor-intensive.

One way of overcoming this problem is to reduce the specificity of senses and focusing on a few dominant senses [Mohammad and Hirst, 2006]. In addition to this, one can maximize the use of knowledge one gathers from available labeled data, by identifying common 'classes' of word senses depending on the similarities in their usage.

WordNet *lexicographer files* (**LFs**) are sense groups, that are created manually during the construction of WordNet [Fellbaum, 1998]. They provide a rough classification of senses. For instance, first senses of nouns *cat* and *dog* fall into the LF ANIMAL, and first sense of verb *dance* falls into MOTION. WordNet has 25 LFs defined for nouns, and 15 for verbs. LFs have been an intuitive choice for semantic classes or 'supersenses' for words due to many reasons, including the popularity of WordNet as a lexical resource, and availability of data labeled with respect to WordNet senses. Two works discuss how to use LFs in fine grained WSD: Crestan et al. [2001] classified word instances into WordNet LFs in *Senseval-2* evaluation exercise. Kohomban and Lee [2005] proposed how training examples from different words can be utilized to learn WordNet LFs, which could then be used for fine-grained WSD of nouns and verbs. Ciaramita and Johnson [2003] used contextual features to classify unknown nouns into WordNet LFs.

This use of WordNet LFs begs the question: can we do better if we design the sense classes specifically for fine-grained WSD? We answer this question in the affirmative, by using clustering techniques to automatically derive the sense classes. We show that sense classes constructed in this way significantly outperform the WordNet LFs when used with Kohomban and Lee's [2005] method for all word fine-grained WSD. One interesting result is that the amount of inevitable losses, caused by multiple fine-grained senses falling into the same sense class, can be made dramatically smaller, even when the number of sense classes is kept the same as the number of WordNet LFs. Additionally, our method can be applied to parts of speech other than nouns and verbs, where WordNet LFs cannot be effectively used. The resulting WSD system yields state of the art results on the *Senseval-2* and *3* English all-words task evaluation datasets; our result on *Senseval-3* data is the best that we are aware of.

1.1 Generic Word Sense Classes: Motivation

This work borrows from [Kohomban and Lee, 2005; Crestan *et al.*, 2001] one major idea: if we can classify word instances into a coarse-grained set of word sense *classes*, and if we know which fine-grained senses fall into these classes, then we can always use the same system as a fine-grained WSD system by replacing the resulting coarse-grained classes with the most frequent fine-grained sense within each class. This way we lose some senses, hence some accuracy. Kohomban and Lee [2005] argued that this loss can be affordable, given that the classifier can gain from coarser granularity, as coarser classes reduce the data sparsity. Our results from this paper show that with properly designed classes, this loss can be made far smaller than the loss of the WordNet LFs previously used.

A system working on this principle uses the fine-to-coarse sense mapping to convert any available data, labeled with fine-grained senses, into training examples for coarse-grained classes. A classifier uses training examples from *different words*, to label any word instance into a class containing one or more of its senses; fine grained sense is then assigned in the manner described above. All one needs for this scheme to work is a mapping of fine-grained senses to a coarse set, generic for all words; most previous work used WordNet LFs.

However, WordNet LFs are not designed to work as a generic set of classes for WSD. Thinking on the WSD application setting, one can identify two issues that can hinder the WSD performance when LFs are used as sense classes:

Feature Coherence
Commonly used features in WSD are those available *within text*, such as collocations and part of speech. To the best of our knowledge, there is no proven evidence that WordNet LFs form cohesive classes in term of these features; counter examples can be found.

Wierzbicka [1996], for instance, provides examples that show that even closely related word/hypernym pairs do not share same usage patterns: Word pairs such as *apple/fruit*, *fish/animal*, *insect/animal*, are not readily interchangeable in practical language usage although they are cohesive parts of taxonomy; *"there is an animal on your collar"* sounds very odd although insect is an animal in WordNet terms. Where the linguistic usage is much different, assuming all examples to be in the same class would merely introduce noise. On the other hand, contextually similar usages could have been put into further-away WordNet taxonomies for semantic reasons, making it impractical to differentiate those senses using contextual features alone. Also some semantically close word senses are assigned totally different LFs; for instance *ionosphere/1*:LOCATION and *stratosphere/1*:OBJECT.

Another problem with WordNet LFs is that some LFs are subsumed by others: FOOD, for instance, is a subset of SUBSTANCE. This can create confusion in features when learning. Also some LFs with arguably close meanings, such as COGNITION, FEELING, and MOTIVE, may be hard to differentiate using contextual features alone. It might possibly be better to group them into a single class.

LFs for adjectives do not relate to either the underlying concept or contextual features, and are not applicable as generic semantic classes.

Loss of Senses
The coarser the classes, the greater the chance that a given class includes more than one sense of a given word. As fine-sense to class mapping (described at the beginning of this section) is a many-to-one mapping, we can lose a few senses for each word in the reverse mapping, resulting in errors in fine-grained WSD. Granularity of senses that a fine-grained WSD system can attain with WordNet LFs is poor, having only 25 classes for nouns and 15 for verbs.

1.2 Clustering as a Solution

In order to address these two issues, we suggest a more direct, *task-oriented* approach.

Using the features within text to find common groups of senses based on their context has been shown to be useful previously [Lin and Pantel, 2002; Magnini and Cavaglià, 2000]. We use this idea for generic sense classes: using clustering techniques, we try to generate automatically a set of 'classes' of word senses that are based on lexical and syntactic features alone. Since these classes are directly based upon features, unlike WordNet LFs, we expect them to be easier to learn using the same features. There is no new *linguistic* assumption made here; the only assumption made on classes is that the senses that fall into the same class, by showing similar usage patterns in a labeled corpus, will show consistent behavior elsewhere. This is the basic reasoning behind inductive learning.

We address the issue of sense-loss due to coarse grain nature of WordNet LFs by having a larger number of classes than WordNet does as LFs.

It could be argued that the WordNet hierarchy encodes much of human hand-crafted knowledge, and should be retained as much as possible in the construction of coarse-grained classes. We tested this idea by partitioning the WordNet hierarchy into segments that are finer than the WordNet LFs, while retaining the WordNet hierarchical relationships within each partition. Our result shows that this method does not work well, both for reducing sense loss as well as in the final classifier performance.

In the next section, we will describe these two clustering schemes, i.e. purely feature based and WordNet-hierarchy constrained. Section 3 will analyze how well we managed to reach our design goals of feature coherence and sense granularity, using feature based sense clustering. In section 4, we present the framework we set up for evaluating the performance of the classes in the real *end-task*: fine grained WSD. Section 5 discusses the results, and we show that improvement over state of the art is possible with our system, comparing with previously published results.

2 Clustering Schemes

This section describes the implementation of our proposal, automatic generation of classes based on features, and the control experiment, where we used similar techniques to obtain classes that are constrained within WordNet hierarchies. These will be referred to as **FB** and **WN** respectively.

Syntactic and lexical features in text do not necessarily correlate. For this reason, we decided to test two different clustering arrangements, which are respectively based on local context and Part of Speech features from labeled data (See sections 4.1 and 4.2 for details on data and features). Features are represented as a binary vector. Local context feature vectors were of large dimension, but were very sparse; we used singular value decomposition to reduce feature space dimension, and discard elements with singular values smaller than 1% of the largest. Data thus obtained is used in FB and WN schemes. Each scheme has two class arrangements, based on local context and POS features.

2.1 Purely Feature-Based Classes (FB)

In this section, we discuss clustering senses independently of the original WordNet hierarchy; our target here is better feature-class coherence. The idea is that as long as the corpus behavior of two senses remain the same, it is possible to assume them as being in some hypothetical generic 'class', regardless of our being able to understand, or label, the exact semantics of that class. If we can find such classes using labeled data, then it must be possible to use them in WSD, in place of WordNet LFs, as described in section 1.1.

A sense is represented by the average of vectors of labeled instances in the corpus for that sense. We omitted the senses

that are absent in the labeled corpus, and in the WSD task (section 4), considered them to have their own classes.

Our clustering algorithm is inspired by the k-means+ algorithm [Guan *et al.*, 2004]. Instead of initializing the clusters randomly, we chose to base them on original WordNet LFs. Instead of iterating with a fixed number of clusters, we used a method of growing new clusters from outliers of existing clusters. After each iteration of k-means algorithm, we calculate the variance of clusters formed. Then we check the squared distance of each point in the cluster to its centroid; if the ratio of this distance to variance is larger than a given constant,[1] the point is isolated as a new cluster on its own. Upon reaching convergence, another refinement is made: if a cluster has a smaller number of members than desirable, we merge it with the nearest cluster, chosen by simple-linkage condition: that is, cluster c_j is the 'nearest' to cluster c_i ($i \neq j$) if c_j has the node within the shortest possible distance to any node in c_i. This allows for non-spherical clusters, while preserving the size of clusters above a certain limit.

Once the clusters are formed, it is straightforward to create the sense mapping, which can be used in our classifier as discussed above (also in section 4.3). We applied this method for nouns, verbs and adjectives.[2]

2.2 Classes Constrained within WordNet (WN)

First, we build trees from WordNet hierarchy, with senses as nodes, and their hypernyms as respective parent nodes. Trees that belong to same LF are connected together with a root node. Then, the feature coordinates (as earlier) of each sense are added to the tree at its respective node. For a given tree segment, the centroid of coordinates can be calculated by averaging all sense coordinates within that segment; average square distance to centroid is a measure of cohesiveness of a tree. We consider each node in the tree as a candidate breaking point, and decide where to break by checking which split gives the largest reduction in total variance of the system. The partitioning proceeds in a greedy manner, by selecting at each run the node that gives the best overall improvement.

As earlier, smaller clusters were removed by merging them back; however, we cannot pick the geometrically nearest cluster to merge as this would distort the WordNet hierarchy consistency requirement. So a cluster was merged back to the point from which it was originally detached.

Adjectives and adverbs cannot be organized into proper tree forms as they do not have hypernyms. So this method was limited to nouns and verbs only.

3 Effects of Clustering

In this section, we will analyze empirically the basic effects of our clustering schemes, discussing how effectively we managed to obtain the properties we desired as design goals.

[1]This constant was chosen to be slightly below the maximum distance/variance ratio found after the first iteration.

[2]Lexical file arrangement of adjectives is not semantically based; clusters were still initialized with the three available.

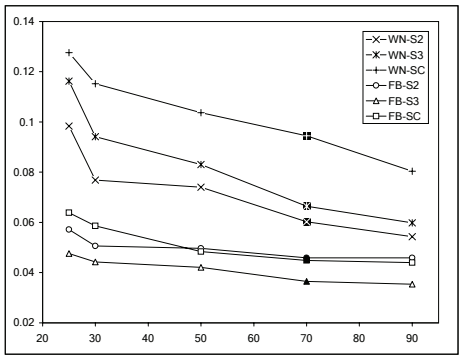

Figure 1: Proportional 'loss' of senses vs number of classes for nouns, for FB and WN: S2, S3, SC are *Senseval-2, 3* all-words task data and SemCor; optimal clustering is highlighted. Left-most points of WN correspond to WordNet LFs. Feature-based classes consistently yield better sense-separation, even at smaller numbers of classes.

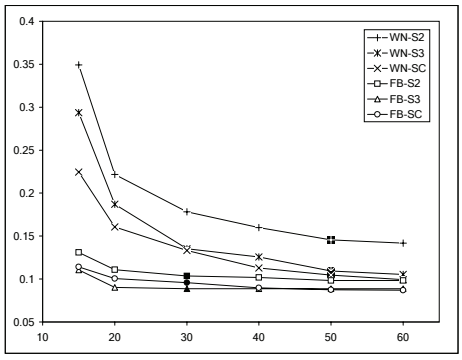

Figure 2: Proportional 'loss' of senses vs number of classes for verbs. Details as per figure 1.

3.1 Sense Resolution

We compared the 'sense loss' of the FB classes with that of the WN classes.

Recall from section 1.1 how the sense loss occurs; losses will be counted as errors in the fine-grained WSD system, so minimizing losses is a desirable quality of classes. Given a class mapping and a labeled corpus, we can assess the loss by counting the proportion of labeled instances that belong to 'lost' senses. Figure 1 and 2 shows this loss in labeled data sets due to FB and WN, at different numbers of classes. WN starting points in the graph are original WordNet LFs.

Although both schemes seem to benefit from larger numbers of classes, there is an additional gain in FB that we did not anticipate: it achieves good resolution, even at smaller numbers of classes. At the same number of classes as the WordNet LFs, it is possible to obtain more than a 50% reduction in sense loss. Recall that FB can either split clusters or reorganize points in order to reduce variance, while WN can only split, as reorganizing would violate the hierarchy. This means that FB can, in theory, achieve better optimization for the same number of clusters, and this seems to work in practice as well. This is an added advantage, as smaller clusters,

	local context		POS	
	nouns	verbs	nouns	verbs
WordNet LFs	0.251	0.223	0.011	0.021
WN	0.336	0.330	0.041	0.065
FB	0.352	0.335	0.065	0.072

Table 1: Average information gain values

although reducing the sense loss, have the undesirable property of including a fewer number of senses. This limits the number of training examples per class.

In the actual WSD task, we used cross validation to guess the best number of classes for each clustering; these are shown highlighted in the graphs.

3.2 Feature Coherence

It was observed that a given FB class can include groupings of different semantics. For instance, a small noun class was dominated by three distinct 'themes' - some of them were {*kneecap/1*,*forearm/1*, *palm/1*}, {*coattail/1*, *overcoat/1*, *shirtsleeve/1*}, and {*homeland/1*, *motherland/1*}. But this mix does not pose a problem as similarity weighting of instances (see section 4.3) can lessen the influence from unrelated words as training instances. A similarity measure based on WordNet hierarchical proximity introduces some of the hierarchy information back to the classification, ensuring both contextual and taxonomical coherence.

To empirically evaluate how well these classes can be separated by features in the end-task classifier, we calculated feature information gain values for 6-token POS/local context window on complete SemCor data set. Information gain of a feature i with set of values V_i is given by

$$w_i = H(C) - \sum_{v \in V_i} P(v) \cdot H(C|v),$$

where $H(C) = -\sum_{c \in C} P(c) \log P(c)$ is the entropy of the class distribution C. This provides a measure of how well a given set of classes can be separated by a given feature. Since the class distribution C and feature values v for each feature are available for SemCor, it is straightforward to apply the formula to obtain w_i for a given feature. Table 1 shows information gain measures in nouns and verbs for SemCor data, for local context and POS features (averaged over six positions in context windows). It can be seen that both WN and FB clusters improve the gain with smaller class sizes, but FB clusters yield the best gain.

4 WSD End-Task Evaluation Framework

In order to empirically validate the effect of the two properties of classes, which we thought of as critical for WSD end-task, we used the system originally described in [Kohomban and Lee, 2005] for fine-grained WSD.[3] We measure the 'quality' of our classes by using them instead of WordNet LFs that were used in the original work, and evaluating the performance of the resulting WSD system.

4.1 Data

We use labeled data from SemCor corpus [Fellbaum, 1998, chapter 8] as training data. To determine global classifier settings, a randomly selected part of this (1000 instances for each part of speech) is held-out as validation data set. Where word-level validation is employed, randomly picked word instances (up to 20) from the training data were kept aside. Evaluation was done on *Senseval-2* and *Senseval-3* [Edmonds and Cotton, 2001; Snyder and Palmer, 2004] English all-words task data sets. Our tests use WordNet 1.7.1 senses.

4.2 Features

Features used for learning are implemented in the same way as described in [Kohomban and Lee, 2005].

Local context: This is a $[-n, +n]$ symmetric window of words to the both sides of word under consideration. The size of the window $n \in \{1, 2, 3\}$ was chosen by cross validation. All words were converted to lower case and punctuation marks were excluded.

Part of speech: This is similar to the local context window, using the same rules of not considering parts of speech of punctuations and not exceeding sentence boundaries.

Grammatical relations: This included the basic syntactic relations such as subject-verb and verb-object, as well as prepositional phrases, such as *'sound of bells'* for word *sound*.

4.3 Classifier

All we obtain from the clustering process is the mapping from fine grained senses into their respective class number. Each fine-grained sense labeled instance in the training set is used in the classifier as an example for that particular class, using the class mapping. Training data for each word is limited to those instances belonging to classes that include one or more senses of that word. The classifier used is TiMBL memory based learner [Daelemans *et al.*, 2003] which is essentially a k-NN classifier.

Kohomban and Lee [2005] showed that the noise due to using examples from different words could be reduced if the examples were weighted according to the relatedness of example-instance word to the word being labeled. We employ the same method for nouns and verbs in our experiment, using the same relatedness measure for weighting, proposed by Jiang and Conrath (**JCn**) [1997]. This measure takes in to account the proximity of the two senses within the WordNet hierarchy, as well as the information content of the nodes of the path which links them together.[4]

Instance weighting is implemented by modifying the distance $\Delta(X, Y)$ between a training instance X and testing instance Y, in the following way:

$$\Delta^E(X, Y) = \frac{\Delta(X, Y)}{S_{X,Y} + \epsilon}$$

[3] Some improvements were made in the system after [Kohomban and Lee, 2005] was published, so we re-ran the experiments reported there, which used WordNet LFs as sense classes. The new results correspond to 'WordNet LFs' entries in the tables.

[4] Our implementation uses information content data compiled by Ted Pedersen et al. http://wn-similarity.sourceforge.net/.

$S_{X,Y}$ is the JCn similarity between X and the most frequent sense of Y that falls within the class of X; ϵ is a small constant to avoid division by zero.

A separate classifier was used for each of the three feature type described above. These three classifiers, and a classifier which always predicts WordNet first sense as its output, participated in simple majority voting. In case of a tie, the first sense was chosen.

Weighted Majority Voting As Kohomban and Lee [2005] reported, Weighted majority algorithm [Littlestone and Warmuth, 1994] can increase classifier performance over simple majority algorithm. For the final combination, we used the validation data to pick the best clustering scheme (POS or local context) as well. Nouns and adjectives did well with local context based clusters, while verbs did well on POS based clusters.

Final Results

Once an instance is classified as a particular class, we have to decide which fine-grained sense it belongs to. There can be more than one sense that falls into a given class. As mentioned in section 1.1, we use the heuristic of picking the sense with the smallest WordNet sense number within the class. This is motivated by the fact that WordNet senses supposedly come in descending order of their frequency.

For each clustering, the words with senses that fall into multiple classes were classified as described in this section. All the other words are assigned WordNet sense 1. Also, we check for multi-word phrases that have separate WordNet entries. These phrases were assigned sense 1 as they usually have only one sense. Adjective clustering was not applicable for WN, as mentioned earlier. In FB, method used for adjectives was the same as above; in WN, and for adverbs in both cases, we resorted to WordNet first sense.

Adjective Similarity

JCn similarity measure depends on hypernym hierarchy, and is not applicable for adjectives, which do not have a hierarchical organization. As Kohomban and Lee [2005] reported, only JCn similarity measure could help the classifiers outperform the baseline. In addition, the adjective LFs, only three in number, are not suitable for the WSD framework. However, FB clustering scheme could be applied on adjectives as well, as we can group adjective senses into smaller classes. In addition, the context vectors we get from SVD provide a way for calculating inter-sense similarity. The coordinate vectors resulting from SVD gives a smoothed out measure of average behavior of a sense. This idea has been successfully used in WSD previously [Strapparava *et al.*, 2004]. We could use it as a measure of similarity as well, by using the dot product of coordinate vectors as the similarity between senses.

In general, when we used this measure in the classifier process, it yielded results that outperformed the baseline. However the measure could not outperform JCn. So we limited its use to adjectives only.

5 Results

We evaluated the two clustering schemes FB and WN in the framework described in section 4, in clustering schemes that are based on both POS and local context features.

	Senseval-2	*Senseval-3*
Baseline	0.658	0.643
Senseval Best	0.690	0.652
Crestan et al.	0.618	-

Table 2: Baseline and previous results

	noun	**verb**	**adj.**	**combined**
baseline	0.711	0.439	0.639	0.658
WordNet LFs	0.724	0.455	0.639	0.668
WN, POS	0.724	0.453	0.639	0.667
WN, LC	0.723	0.457	0.639	0.667
FB, POS	0.725	**0.480**	0.643	0.674
FB, LC	**0.747**	0.458	**0.654**	**0.681**
baseline	0.700	0.534	0.669	0.643
WordNet LFs	0.719	0.548	0.669	0.656
WN, POS	0.717	0.559	0.669	0.658
WN, LC	0.719	0.557	0.669	0.659
FB, POS	0.710	**0.568**	0.694	0.664
FB, LC	**0.736**	0.541	**0.708**	**0.668**

Table 3: Results for different original WordNet LFs, WN and FB clustering schemes, simple majority voting, for *Senseval-2* (above) and *Senseval-3* (below) data. POS and LC are clusterings based on POS and local context features (section 2).

Table 2 shows the baseline (WordNet first sense) and the performance[5] of two best systems reported in *Senseval*, [Mihalcea, 2002] and [Decadt *et al.*, 2004], as well as that of [Crestan *et al.*, 2001], which used WordNet LFs.

Table 3 shows the results of the clustering schemes we discussed, using simple majority voting (see section 4.3), as well as the re-run of the system with original WordNet LFs. Results are given for three parts of speech and the combined system (as described in 'Final Results'). WN clusters' performance is not significantly different from that of original WordNet LFs; this may be because the constraining on the hierarchy had the same type of localization effect that original system achieved with JCn weighting, thus not yielding any additional information and even preventing some informative examples from being used. This supports our idea that one cannot obtain good performance merely by splitting LF into a finer set of classes.

On the other hand, the feature-based (FB) classes provide *contextually-based* information that are not given by the hierarchy, and could have complemented the information from the JCn similarity measure, which is based on the hierarchy. In other words, clustering independently of the hierarchy is a better way to utilize the two different sources of information: semantic (from taxonomical hierarchy), and lexical/syntactic (from linguistic usage patterns).

It can also be seen that for FB, POS based clustering performed well for verbs, while local context based clustering did well with nouns and adjectives. A rough explanation may be that verbs generally benefit from syntactic features, which are available through POS. The effect is not consistent for

[5] all numbers shown are recall values using the official scorer.

	Senseval-2	*Senseval-3*
WordNet LFs	0.674	0.661
WN	0.664	0.659
FB	**0.687**	**0.677**

Table 4: Results after weighted majority voting.

WN; again, this may be due to the fact that hierarchical constraining impedes the clustering effectiveness as well as the utility of JCn.

Table 4 show the results after using weighted majority algorithm for classifier combination. The results of the feature based classes here is better than all other systems. Except for [Mihalcea, 2002], it outperforms all previously reported state-of-the art results in respective *Senseval* tasks.

Compared with the results using WordNet LFs, improvements given by our feature based system (on complete data set) is statistically significant on McNemar test ($p < 0.01$). Although the improvements over previous systems are numerically small, the figures are considerable when compared with the similar order improvements of the state-of-the-art systems over baseline performance.

6 Conclusion

We explored the idea of using generic word sense classes for fine-grained WSD, and discussed some issues one faces when using WordNet lexicographer files as sense classes. We proposed an alternative *task-oriented* classification scheme, a set of generic sense classes that are based on lexical and syntactic features of text. We gain better classifier accuracy by optimizing the set of target classes to suit the system, instead of the common practice of optimizing the classifier or features. In addition, our system can be used on WSD of parts of speech such as adjectives, where WordNet lexicographer files are inapplicable.

We evaluated the classes we generated by implementing a system that previously reported good results on WSD by learning WordNet LFs, and using the classes we generated in place of the LFs. Our results show that the new classes can improve over WordNet LFs, and yield results that outperform most of the best results on *Senseval-2*, and the best published results on *Senseval-3*.

References

[Ciaramita and Johnson, 2003] M. Ciaramita and M. Johnson. Supersense tagging of unknown nouns in wordnet. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2003.

[Crestan *et al.*, 2001] Eric Crestan, Marc El-Bèze, and C. De Loupy. Improving WSD with multi-level view of context monitored by similarity measure. In *Proceeding of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*, 2001.

[Daelemans *et al.*, 2003] Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. TiMBL: Tilburg memory based learner, version 5.1. Technical report, ILK 03-10, 2003.

[Decadt *et al.*, 2004] Bart Decadt, Véronique Hoste, Walter Daelemans, and Antal Van den Bosch. GAMBL, genetic algorithm optimization of memory-based WSD. In *Senseval-3: Third Intl. Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004.

[Edmonds and Cotton, 2001] Phil Edmonds and Scott Cotton. Senseval-2: Overview. In *Proc. of the Second Intl. Workshop on Evaluating Word Sense Disambiguation Systems (Senseval-2)*, 2001.

[Fellbaum, 1998] Christine Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA, 1998.

[Guan *et al.*, 2004] Yu Guan, Ali A. Ghorbani, and Nabil Belacel. K-means+: An autonomous clustering algorithm. Technical Report TR04-164, University of New Brunswick, 2004.

[Jiang and Conrath, 1997] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, Taiwan, 1997.

[Kohomban and Lee, 2005] Upali S. Kohomban and Wee Sun Lee. Learning semantic classes for word sense disambiguation. In *Proc. of the 43rd Annual Meeting of the Assoc. for Comp. Linguistics (ACL'05)*, June 2005.

[Lin and Pantel, 2002] Dekang Lin and Patrick Pantel. Concept discovery from text. In *Proc. of the 19th intl. conf. on Computational linguistics*, pages 1–7, 2002.

[Littlestone and Warmuth, 1994] N Littlestone and M.K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[Magnini and Cavaglià, 2000] B. Magnini and G. Cavaglià. Integrating subject field codes into WordNet. In *Proc. of LREC-2000, Second Intl. Conf. on Language Resources and Evaluation*, 2000.

[Mihalcea, 2002] R. Mihalcea. Word sense disambiguation using pattern learning and automatic feature selection. In *Jnl. of Nat. Language and Engineering*, December 2002.

[Mohammad and Hirst, 2006] Saif Mohammad and Graeme Hirst. Determining word sense dominance using a thesaurus. In *Proc. of the 11th Conf. of the European Chapter of the Assoc. for Computational Linguistics*, 2006.

[Snyder and Palmer, 2004] Benjamin Snyder and Martha Palmer. The English all-words task. In *Senseval-3: Third Intl. Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004.

[Strapparava *et al.*, 2004] C. Strapparava, A. Gliozzo, and C. Giuliano. Pattern abstraction and term similarity for word sense disambiguation: IRST at Senseval-3. In *Senseval-3: 3rd Intal. Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004.

[Wierzbicka, 1996] Anna Wierzbicka. Semantics and Ethnobiology. In *Semantics: primes and universals*, pages 351–376. Oxford University Press, 1996.