Example-Based Metonymy Recognition for Proper Nouns

Yves Peirsman
Quantitative Lexicology and Variational Linguistics
University of Leuven, Belgium
yves.peirsman@arts.kuleuven.be

Abstract

Metonymy recognition is generally approached with complex algorithms that rely heavily on the manual annotation of training and test data. This paper will relieve this complexity in two ways. First, it will show that the results of the current learning algorithms can be replicated by the 'lazy' algorithm of Memory-Based Learning. This approach simply stores all training instances to its memory and classifies a test instance by comparing it to all training examples. Second, this paper will argue that the number of labelled training examples that is currently used in the literature can be reduced drastically. This finding can help relieve the knowledge acquisition bottleneck in metonymy recognition, and allow the algorithms to be applied on a wider scale.

1 Introduction

Metonymy is a figure of speech that uses "one entity to refer to another that is related to it" (Lakoff and Johnson, 1980, p.35). In example (1), for instance, *China* and *Taiwan* stand for the governments of the respective countries:

(1) *China* has always threatened to use force if *Taiwan* declared independence. (BNC)

Metonymy resolution is the task of automatically recognizing these words and determining their referent. It is therefore generally split up into two phases: metonymy recognition and metonymy interpretation (Fass, 1997).

The earliest approaches to metonymy recognition identify a word as metonymical when it violates selectional restrictions (Pustejovsky, 1995). Indeed, in example (1), *China* and *Taiwan* both violate the restriction that *threaten* and *declare* require an animate subject, and thus have to be interpreted metonymically. However, it is clear that many metonymies escape this characterization. *Nixon* in example (2) does not violate the selectional restrictions of the verb *to bomb*, and yet, it metonymically refers to the army under Nixon's command.

(2) *Nixon* bombed Hanoi.

This example shows that metonymy recognition should not be based on rigid rules, but rather on statistical information about the semantic and grammatical context in which the target word occurs.

This statistical dependency between the reading of a word and its grammatical and semantic context was investigated by Markert and Nissim (2002a) and Nissim and Markert (2003; 2005). The key to their approach was the insight that metonymy recognition is basically a subproblem of Word Sense Disambiguation (WSD). Possibly metonymical words are polysemous, and they generally belong to one of a number of predefined metonymical categories. Hence, like WSD, metonymy recognition boils down to the automatic assignment of a sense label to a polysemous word. This insight thus implied that all machine learning approaches to WSD can also be applied to metonymy recognition.

There are, however, two differences between metonymy recognition and WSD. First, theoretically speaking, the set of possible readings of a metonymical word is open-ended (Nunberg, 1978). In practice, however, metonymies tend to stick to a small number of patterns, and their labels can thus be defined a priori. Second, classic

WSD algorithms take training instances of one particular word as their input and then disambiguate test instances of the same word. By contrast, since all words of the same semantic class may undergo the same metonymical shifts, metonymy recognition systems can be built for an entire semantic class instead of one particular word (Markert and Nissim, 2002a).

To this goal, Markert and Nissim extracted from the BNC a corpus of possibly metonymical words from two categories: country names (Markert and Nissim, 2002b) and organization names (Nissim and Markert, 2005). All these words were annotated with a semantic label — either `literal` or the metonymical category they belonged to. For the country names, Markert and Nissim distinguished between `place-for-people`, `place-for-event` and `place-for-product`. For the organization names, the most frequent metonymies are `organization-for-members` and `organization-for-product`. In addition, Markert and Nissim used a label `mixed` for examples that had two readings, and `othermet` for examples that did not belong to any of the pre-defined metonymical patterns.

For both categories, the results were promising. The best algorithms returned an accuracy of 87% for the countries and of 76% for the organizations. Grammatical features, which gave the function of a possibly metonymical word and its head, proved indispensable for the accurate recognition of metonymies, but led to extremely low recall values, due to data sparseness. Therefore Nissim and Markert (2003) developed an algorithm that also relied on semantic information, and tested it on the mixed country data. This algorithm used Dekang Lin's (1998) thesaurus of semantically similar words in order to search the training data for instances whose head was similar, and not just identical, to the test instances. Nissim and Markert (2003) showed that a combination of semantic and grammatical information gave the most promising results (87%).

However, Nissim and Markert's (2003) approach has two major disadvantages. The first of these is its complexity: the best-performing algorithm requires smoothing, backing-off to grammatical roles, iterative searches through clusters of semantically similar words, etc. In section 2, I will therefore investigate if a metonymy recognition algorithm needs to be that computationally demanding. In particular, I will try and replicate Nissim and Markert's results with the 'lazy' algorithm of Memory-Based Learning.

The second disadvantage of Nissim and Markert's (2003) algorithms is their supervised nature. Because they rely so heavily on the manual annotation of training and test data, an extension of the classifiers to more metonymical patterns is extremely problematic. Yet, such an extension is essential for many tasks throughout the field of Natural Language Processing, particularly Machine Translation. This knowledge acquisition bottleneck is a well-known problem in NLP, and many approaches have been developed to address it. One of these is active learning, or sample selection, a strategy that makes it possible to selectively annotate those examples that are most helpful to the classifier. It has previously been applied to NLP tasks such as parsing (Hwa, 2002; Osborne and Baldridge, 2004) and Word Sense Disambiguation (Fujii et al., 1998). In section 3, I will introduce active learning into the field of metonymy recognition.

2 Example-based metonymy recognition

As I have argued, Nissim and Markert's (2003) approach to metonymy recognition is quite complex. I therefore wanted to see if this complexity can be dispensed with, and if it can be replaced with the much more simple algorithm of Memory-Based Learning. The advantages of Memory-Based Learning (MBL), which is implemented in the TiMBL classifier (Daelemans et al., 2004)[1], are twofold. First, it is based on a plausible psychological hypothesis of human learning. It holds that people interpret new examples of a phenomenon by comparing them to "stored representations of earlier experiences" (Daelemans et al., 2004, p.19). This contrasts to many other classification algorithms, such as Naive Bayes, whose psychological validity is an object of heavy debate. Second, as a result of this learning hypothesis, an MBL classifier such as TiMBL eschews the formulation of complex rules or the computation of probabilities during its training phase. Instead it stores all training vectors to its memory, together with their labels. In the test phase, it computes the distance between the test vector and all these train-

[1]This software package is freely available and can be downloaded from http://ilk.uvt.nl/software.html.

ing vectors, and simply returns the most frequent label of the most similar training examples.

One of the most important challenges in Memory-Based Learning is adapting the algorithm to one's data. This includes finding a representative seed set as well as determining the right distance measures. For my purposes, however, TiMBL's default settings proved more than satisfactory. TiMBL implements the IB1 and IB2 algorithms that were presented in Aha et al. (1991), but adds a broad choice of distance measures. Its default implementation of the IB1 algorithm, which is called IB1-IG in full (Daelemans and Van den Bosch, 1992), proved most successful in my experiments. It computes the distance between two vectors X and Y by adding up the weighted distances δ between their corresponding feature values x_i and y_i:

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \delta(x_i, y_i) \quad (3)$$

The most important element in this equation is the weight that is given to each feature. In IB1-IG, features are weighted by their Gain Ratio (equation 4), the division of the feature's Information Gain by its split info. Information Gain, the numerator in equation (4), "measures how much information it [feature i] contributes to our knowledge of the correct class label [...] by computing the difference in uncertainty (i.e. entropy) between the situations without and with knowledge of the value of that feature" (Daelemans et al., 2004, p.20). In order not "to overestimate the relevance of features with large numbers of values" (Daelemans et al., 2004, p.21), this Information Gain is then divided by the *split info*, the entropy of the feature values (equation 5). In the following equations, C is the set of class labels, $H(C)$ is the entropy of that set, and V_i is the set of values for feature i.

$$w_i = \frac{H(C) - \sum_{v \in V_i} P(v) \times H(C|v)}{si(i)} \quad (4)$$

$$si(i) = -\sum_{v \in V_i} P(v) log_2 P(v) \quad (5)$$

The IB2 algorithm was developed alongside IB1 in order to reduce storage requirements (Aha et al., 1991). It iteratively saves only those instances that are misclassified by IB1. This is because these will likely lie close to the decision boundary, and hence, be most informative to the classifier. My experiments showed, however, that IB2's best performance lay more than 2% below that of IB1. It will therefore not be treated any further here.

2.1 Experiments with grammatical information only

In order to see if Memory-Based Learning is able to replicate Nissim and Markert's (2003; 2005) results, I used their corpora for a number of experiments. These corpora consist of one set with about 1000 mixed country names, another with 1000 occurrences of *Hungary*, and a final set with about 1000 mixed organization names.[2] Evaluation was performed with ten-fold cross-validation.

The first round of experiments used only grammatical information. The experiments for the location data were similar to Nissim and Markert's (2003), and took the following features into account:

- the grammatical function of the word (subj, obj, iobj, pp, gen, premod, passive subj, other);
- its head;
- the presence of a second head;
- the second head (if present).

The experiments for the organization names used the same features as Nissim and Markert (2005):

- the grammatical function of the word;
- its head;
- its type of determiner (if present) (def, indef, bare, demonst, other);
- its grammatical number (sing, plural);
- its number of grammatical roles (if present).

The number of words in the organization name, which Nissim and Markert used as a sixth and final feature, led to slightly worse results in my experiments and was therefore dropped.

The results of these first experiments clearly beat the baselines of 79.7% (countries) and 63.4% (organizations). Moreover, despite its extremely

[2]This data is publicly available and can be downloaded from http://homepages.inf.ed.ac.uk/mnissim/mascara.

	Acc	P	R	F
TiMBL	86.6%	80.2%	49.5%	61.2%
N&M	87.0%	81.4%	51.0%	62.7%

Table 1: Results for the mixed country data.
TiMBL: my TiMBL results
N&M: Nissim and Markert's (2003) results

	Acc	P	R	F
TiMBL	74.63%	78.65%	55.53%	65.10%
N&M	76.0%	—	—	—

Table 2: Results for the mixed organization data.
TiMBL: my TiMBL results
N&M: Nissim and Markert's (2005) results

simple learning phase, TiMBL is able to replicate the results from Nissim and Markert (2003; 2005). As table 1 shows, accuracy for the mixed country data is almost identical to Nissim and Markert's figure, and precision, recall and F-score for the metonymical class lie only slightly lower.[3] TiMBL's results for the Hungary data were similar, and equally comparable to Markert and Nissim's (Katja Markert, personal communication). Note, moreover, that these results were reached with grammatical information only, whereas Nissim and Markert's (2003) algorithm relied on semantics as well.

Next, table 2 indicates that TiMBL's accuracy for the mixed organization data lies about 1.5% below Nissim and Markert's (2005) figure. This result should be treated with caution, however. First, Nissim and Markert's available organization data had not yet been annotated for grammatical features, and my annotation may slightly differ from theirs. Second, Nissim and Markert used several feature vectors for instances with more than one grammatical role and filtered all `mixed` instances from the training set. A test instance was treated as `mixed` only when its several feature vectors were classified differently. My experiments, in contrast, were similar to those for the location data, in that each instance corresponded to one vector. Hence, the slightly lower performance of TiMBL is probably due to differences between the two experiments.

These first experiments thus demonstrate that Memory-Based Learning can give state-of-the-art performance in metonymy recognition. In this respect, it is important to stress that the results for the country data were reached without any semantic information, whereas Nissim and Markert's (2003) algorithm used Dekang Lin's (1998) clusters of semantically similar words in order to deal with data sparseness. This fact, together with the psychological plausibility and the simple learning phase, adds to the attractivity of Memory-Based Learning.

[3] Precision, recall and F-score are given for the metonymical class only, since this is the category that metonymy recognition is concerned with.

2.2 Experiments with semantic and grammatical information

It is still intuitively true, however, that the interpretation of a possibly metonymical word depends mainly on the semantics of its head. The question is if this information is still able to improve the classifier's performance. I therefore performed a second round of experiments with the location data, in which I also made use of semantic information. In this round, I extracted the hypernym synsets of the head's first sense from WordNet. WordNet's hierarchy of synsets makes it possible to quantify the semantic relatedness of two words: the more hypernyms two words share, the more closely related they are. I therefore used the ten highest hypernyms of the first head as features 5 to 14. For those heads with fewer than ten hypernyms, a copy of their lowest hypernym filled the 'empty' features. As a result, TiMBL would first look for training instances with ten identical hypernyms, then with nine, etc. It would thus compare the test example to the semantically most similar training examples.

However, TiMBL did not perform better with this semantic information. Although F-scores for the metonymical category went up slightly, the system's accuracy hardly changed. This result was not due to the automatic selection of the first (most frequent) WordNet sense. By manually disambiguating all the heads in the training and test set of the country data, I observed that this first sense was indeed often incorrect, but that choosing the correct sense did not lead to a more robust system. Clearly, the classifier did not benefit from WordNet information as Nissim and Markert's (2003) did from Lin's (1998) thesaurus.

The learning curves for the country set allow us to compare the two types of feature vectors

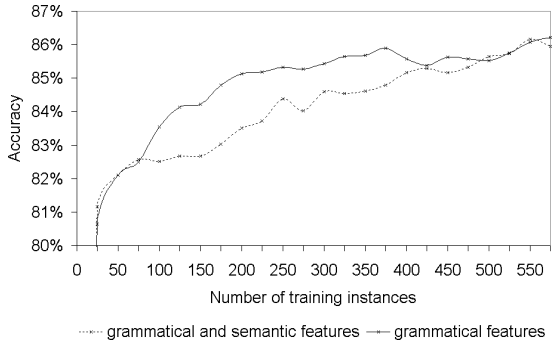

Figure 1: Accuracy learning curves for the mixed country data with and without semantic information.

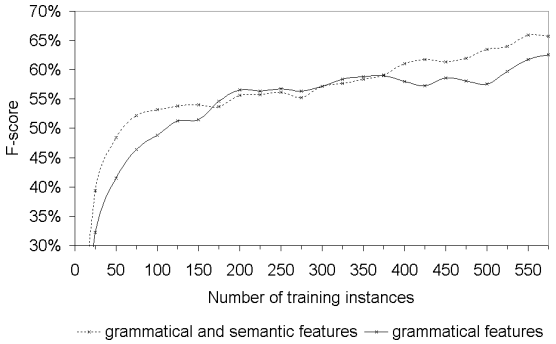

Figure 2: F-score learning curves for the mixed country data with and without semantic information.

in more detail.[4] As figure 1 indicates, with respect to overall accuracy, semantic features have a negative influence: the learning curve with both features climbs much more slowly than that with only grammatical features. Hence, contrary to my expectations, grammatical features seem to allow a better generalization from a limited number of training instances. With respect to the F-score on the metonymical category in figure 2, the differences are much less outspoken. Both features give similar learning curves, but semantic features lead to a higher final F-score. In particular, the use of semantic features results in a lower precision figure, but a higher recall score. Semantic features thus cause the classifier to slightly overgeneralize from the metonymic training examples.

There are two possible reasons for this inability of semantic information to improve the classifier's performance. First, WordNet's synsets do not always map well to one of our semantic labels: many are rather broad and allow for several readings of the target word, while others are too specific to make generalization possible. Second, there is the predominance of prepositional phrases in our data. With their closed set of heads, the number of examples that benefits from semantic information about its head is actually rather small.

Nevertheless, my first round of experiments has indicated that Memory-Based Learning is a simple but robust approach to metonymy recognition. It is able to replace current approaches that need smoothing or iterative searches through a thesaurus, with a simple, distance-based algorithm. Moreover, in contrast to some other successful classifiers, it incorporates a plausible hypothesis of human learning.

[4]These curves were obtained by averaging the results of 10 experiments. They show performance on a test set of 40% of the data, with the other 60% as training data.

3 Distance-based sample selection

The previous section has shown that a simple algorithm that compares test examples to stored training instances is able to produce state-of-the-art results in the field of metonymy recognition. This leads to the question of how many examples we actually need to arrive at this performance. After all, the supervised approach that we explored requires the careful manual annotation of a large number of training instances. This knowledge acquisition bottleneck compromises the extrapolation of this approach to a large number of semantic classes and metonymical patterns. This section will therefore investigate if it is possible to automatically choose informative examples, so that annotation effort can be reduced drastically.

For this round of experiments, two small changes were made. First, since we are focusing on metonymy *recognition*, I replaced all specific metonymical labels with the label `met`, so that only three labels remain: `lit`, `met` and `mixed`. Second, whereas the results in the previous section were obtained with ten-fold cross-validation, I ran these experiments with a training and a test set. On each run, I used a random 60% of the data for training; 40% was set aside for testing. All curves give the average of twenty test runs that use grammatical information only.

In general, sample selection proceeds on the basis of the confidence that the classifier has in its classification. Commonly used metrics are the probability of the most likely label, or the entropy

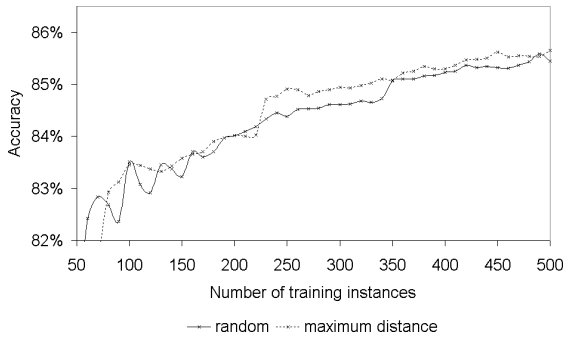

Figure 3: Accuracy learning curves for the country data with random and maximum-distance selection of training examples.

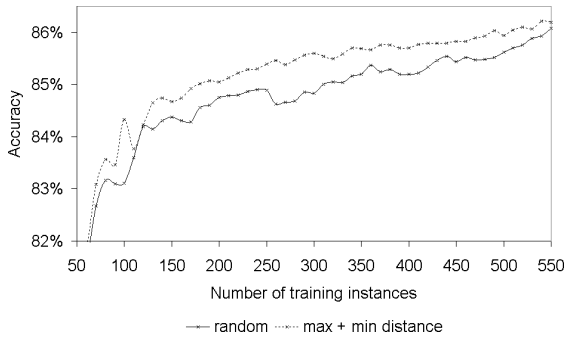

Figure 4: Accuracy learning curves for the country data with random and maximum/minimum-distance selection of training examples.

over all possible labels. The algorithm then picks those instances with the lowest confidence, since these will contain valuable information about the training set (and hopefully also the test set) that is still unknown to the system.

One problem with Memory-Based Learning algorithms is that they do not directly output probabilities. Since they are example-based, they can only give the distances between the unlabelled instance and all labelled training instances. Nevertheless, these distances can be used as a measure of certainty, too: we can assume that the system is most certain about the classification of test instances that lie very close to one or more of its training instances, and less certain about those that are further away. Therefore the selection function that minimizes the probability of the most likely label can intuitively be replaced by one that maximizes the distance from the labelled training instances.

However, figure 3 shows that for the mixed country instances, this function is not an option. Both learning curves give the results of an algorithm that starts with fifty random instances, and then iteratively adds ten new training instances to this initial seed set. The algorithm behind the solid curve chooses these instances randomly, whereas the one behind the dotted line selects those that are most distant from the labelled training examples. In the first half of the learning process, both functions are equally successful; in the second the distance-based function performs better, but only slightly so.

There are two reasons for this bad initial performance of the active learning function. First, it is not able to distinguish between *informative* and *unusual* training instances. This is because a large distance from the seed set simply means that the particular instance's feature values are relatively unknown. This does not necessarily imply that the instance is informative to the classifier, however. After all, it may be so unusual and so badly representative of the training (and test) set that the algorithm had better exclude it — something that is impossible on the basis of distances only. This bias towards outliers is a well-known disadvantage of many simple active learning algorithms. A second type of bias is due to the fact that the data has been annotated with a few features only. More particularly, the present algorithm will keep adding instances whose head is not yet represented in the training set. This entails that it will put off adding instances whose function is `pp`, simply because other functions (`subj`, `gen`, ...) have a wider variety in heads. Again, the result is a labelled set that is not very representative of the entire training set.

There are, however, a few easy ways to increase the number of prototypical examples in the training set. In a second run of experiments, I used an active learning function that added not only those instances that were most distant from the labelled training set, but also those that were closest to it. After a few test runs, I decided to add six distant and four close instances on each iteration. Figure 4 shows that such a function is indeed fairly successful. Because it builds a labelled training set that is more representative of the test set, this algorithm clearly reduces the number of annotated instances that is needed to reach a given performance.

Despite its success, this function is obviously not yet a sophisticated way of selecting good train-

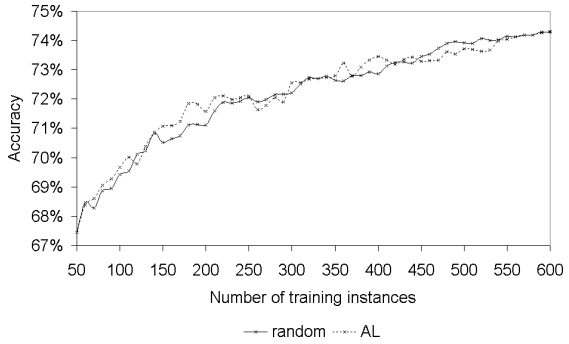

Figure 5: Accuracy learning curves for the organization data with random and distance-based (AL) selection of training examples with a random seed set.

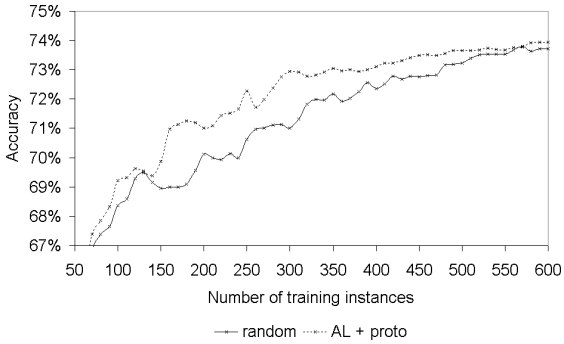

Figure 6: Accuracy learning curves for the organization data with random and distance-based (AL) selection of training examples with a prototypical seed set.

ing examples. The selection of the initial seed set in particular can be improved upon: ideally, this seed set should take into account the overall distribution of the training examples. Currently, the seeds are chosen randomly. This flaw in the algorithm becomes clear if it is applied to another data set: figure 5 shows that it does not outperform random selection on the organization data, for instance.

As I suggested, the selection of prototypical or representative instances as seeds can be used to make the present algorithm more robust. Again, it is possible to use distance measures to do this: before the selection of seed instances, the algorithm can calculate for each unlabelled instance its distance from each of the other unlabelled instances. In this way, it can build a prototypical seed set by selecting those instances with the smallest distance on average. Figure 6 indicates that such an algorithm indeed outperforms random sample selection on the mixed organization data. For the calculation of the initial distances, each feature received the same weight. The algorithm then selected 50 random samples from the 'most prototypical' half of the training set.[5] The other settings were the same as above.

With the present small number of features, however, such a prototypical seed set is not yet always as advantageous as it could be. A few experiments indicated that it did not lead to better performance on the mixed country data, for instance. However, as soon as a wider variety of features is taken into account (as with the organization data), the advantages of a prototypical seed set will definitely become more obvious.

In conclusion, it has become clear that a careful selection of training instances may considerably reduce annotation effort in metonymy recognition. Functions that construct a prototypical seed set and then use MBL's distance measures to select informative as well as typical samples are extremely promising in this respect and can already considerably reduce annotation effort. In order to reach an accuracy of 85% on the country data, for instance, the active learning algorithm above needs 44% fewer training instances than its random competitor (on average). On the organisation data, reduction is typically around 30%. These relatively simple algorithms thus constitute a good basis for the future development of robust active learning techniques for metonymy recognition. I believe in particular that research in this field should go hand in hand with an investigation of new informative features, since the present, limited feature set does not yet allow us to measure the classifier's confidence reliably.

4 Conclusions and future work

In this paper I have explored an example-based approach to metonymy recognition. Memory-Based Learning does away with the complexity of current supervised metonymy recognition algorithms. Even without semantic information, it is able to give state-of-the-art results similar to those in the literature. Moreover, not only is the complexity of current learning algorithms unnecessary; the number of labelled training instances can be reduced drastically, too. I have argued that selective sam-

[5] Of course, the random algorithm in figure 6 still randomly selected its seeds from the *entire* training set.

pling can help choose those instances that are most helpful to the classifier. A few distance-based algorithms were able to drastically reduce the number of training instances that is needed for a given accuracy, both for the country and the organization names.

If current metonymy recognition algorithms are to be used in a system that can recognize all possible metonymical patterns across a broad variety of semantic classes, it is crucial that the required number of labelled training examples be reduced. This paper has taken the first steps along this path and has set out some interesting questions for future research. This research should include the investigation of new features that can make classifiers more robust and allow us to measure their confidence more reliably. This confidence measurement can then also be used in semi-supervised learning algorithms, for instance, where the classifier itself labels the majority of training examples. Only with techniques such as selective sampling and semi-supervised learning can the knowledge acquisition bottleneck in metonymy recognition be addressed.

Acknowledgements

I would like to thank Mirella Lapata, Dirk Geeraerts and Dirk Speelman for their feedback on this project. I am also very grateful to Katja Markert and Malvina Nissim for their helpful information about their research.

References

D. W. Aha, D. Kibler, and M. K. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37–66.

W. Daelemans and A. Van den Bosch. 1992. Generalisation performance of backpropagation learning on a syllabifi cation task. In M. F. J. Drossaers and A. Nijholt, editors, *Proceedings of TWLT3: Connectionism and Natural Language Processing*, pages 27–37, Enschede, The Netherlands.

W. Daelemans, J. Zavrel, K. Van der Sloot, and A. Van den Bosch. 2004. TiMBL: Tilburg Memory-Based Learner. Technical report, Induction of Linguistic Knowledge, Computational Linguistics, Tilburg University.

D. Fass. 1997. *Processing Metaphor and Metonymy*. Stanford, CA: Ablex.

A. Fujii, K. Inui, T. Tokunaga, and H. Tanaka. 1998. Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4):573–597.

R. Hwa. 2002. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276.

G. Lakoff and M. Johnson. 1980. *Metaphors We Live By*. London: The University of Chicago Press.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the International Conference on Machine Learning*, Madison, USA.

K. Markert and M. Nissim. 2002a. Metonymy resolution as a classification task. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, Philadelphia, USA.

K. Markert and M. Nissim. 2002b. Towards a corpus annotated for metonymies: the case of location names. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas, Spain.

M. Nissim and K. Markert. 2003. Syntactic features and word similarity for supervised metonymy resolution. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, Sapporo, Japan.

M. Nissim and K. Markert. 2005. Learning to buy a Renault and talk to BMW: A supervised approach to conventional metonymy. In H. Bunt, editor, *Proceedings of the 6th International Workshop on Computational Semantics*, Tilburg, The Netherlands.

G. Nunberg. 1978. *The Pragmatics of Reference*. Ph.D. thesis, City University of New York.

M. Osborne and J. Baldridge. 2004. Ensemble-based active learning for parse selection. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*. Boston, USA.

J. Pustejovsky. 1995. *The Generative Lexicon*. Cambridge, MA: MIT Press.