Using WordNet Domains In A Supervised Learning Word Sense Disambiguation System

David Bell and Jon Patrick
Sydney Language Technology Research Group
School of Information Technologies
University of Sydney
Sydney, Australia
{dbell,jonpat}@it.usyd.edu.au

Abstract

This paper describes the impact of introducing domain information, obtained from context words, into the process of supervised learning for word sense disambiguation. A word sense disambiguation system is described in which many features can be combined to create a rich feature extraction system. Two Wordnet Domain feature extractors are created for this system and are tested with a combination of other context features. A comparison is made between real and binary valued domain feature vectors. Support vector machines are used as a machine learning tool and two different kernels with various parameters are compared in order to find the ideal learning model for this task and data. Results obtained over the Senseval 2 test data suggest that Wordnet domains perform relatively well as single features but when combined with other context features provide little benefit. It is argued other authors obtain superior results as their methods are tuned to the data set.

1 Introduction

Word sense disambiguation (WSD) is the task of assigning the most appropriate meaning, or sense, to a polysemous word in a particular context. The fields of machine translation, document classification, knowledge acquisition and many others all benefit from having information about the meaning of a particular word, and word sense disambiguation greatly increases the accuracy of the process of associating meaning with these words.

The best results in WSD, particularly over a limited sample of words, have been achieved using supervised learning techniques. These techniques involve the extraction of features from the context of the target word in order to build a feature vector that is then used to create a machine learning classifier. Supervised learning WSD systems usually build a collection of "Word Experts" that specialise in classifying one particular word into its different sense classes.

Supervised learning systems require the construction of a feature vector for each example to be classified. This vector represents the chosen features for that particular example. This vector is then compared with those of training examples to decide on the classification according to a learnt model. The method of this comparison depends on the particular machine learning technique.

Wordnet Domains were first introduced for word sense disambiguation by (Magnini et al., 2002) as part of the Senseval 2 word sense disambiguation task. The system used a technique in which a series of domain vectors were built of length equal to the number of domains to be considered. In this case all 43 domains in the WordNet domains database were considered. The series of vectors consisted of one text vector and several sense vectors. The text vector is a domain vector extracted from the context window around a targeted ambiguous word. Given a set of domains $D = \{D_1, D_2, ..., D_n\}$, a text T and a word at position p, the text vector T_p will be the n-dimensional vector such that the component i is the relevance of D_i for T at the position p. A sense vector s is a domain vector extracted from a word sense. Its length represents the frequency of the occurrence of that sense, and its direction represents the 'mean' vector of the texts where the sense usually occurs. A sense vector is built as the sum of the text vectors of the contexts in the training set containing the sense. The disambiguation procedure of a word occurrence T_p consists of a simple comparison between the text vector T_p and the sense vectors of the word itself. This is done using a dot product between T_p and each sense vector. The result is a ranked list of sense vectors for T_p, a sense is then selected by applying a cutoff. This method has the advantage of

setting the cutoff to decide when a good match occurs. This allows this system to be adjusted to boost precision at the cost of lower recall. The drawback of this system is that it does not allow for any other semantic or syntactic information to be included in the disambiguation decision. Also this system does not allow for other more sophisticated machine learning techniques to be applied such as support vector machines, or maximum entropy. This paper demonstrates a system in which a feature vector is created for each example of a word, which holds domain information for the context of that word and can also hold other information about that example. This feature vector can then be used for many machine learning techniques and allows for experimentation into combining domain information with other features for the purpose of word sense disambiguation.

2 Wordnet Domains

Wordnet Domains is an extension to Wordnet by (Magnini et al., 2002) widening the coverage of domain labels in the existing lexicon. In Wordnet Domains, Wordnet synsets have been annotated with one or more domain labels. A domain may include synsets of different syntactic categories and may also include senses from different sub-hierachies. Additionally domains can group senses of the same word into thematic categories which helps to reduce ambiguity.For example the common WSD example word, **bank**, has ten different senses in WordNet. These senses are shown in Table 2 with corresponding domains.

The WordNet Domains database is broken into a hierachical structure a sample of which is given in Table 1. Table 2 demonstrates the grouping of senses into thematic groups by domain. i.e. senses #2 and #7 which have similar meanings are grouped by the domains GEOGRAPHY and GEOLOGY.

2.1 One domain per discourse

The theory behind using WordNet domains for WSD is based upon the idea of one domain per discourse (ODD). This phenomena is in turn based upon the idea of one sense per discourse (Gale et al., 1992) and was verified by (Magnini et al., 2002). The ODD hypothesis claims that multiple uses of a word in a coherent portion of text tend to share the same domain. Hence using the ODD a word could be disambiguated by

Table 1: A sample of the WordNet Domains Hierarchy

doctrines	archaeology astrology history linguistics literature philosophy psychology art religion
art	dance drawing music photography plastic arts theatre
drawing	painting philately

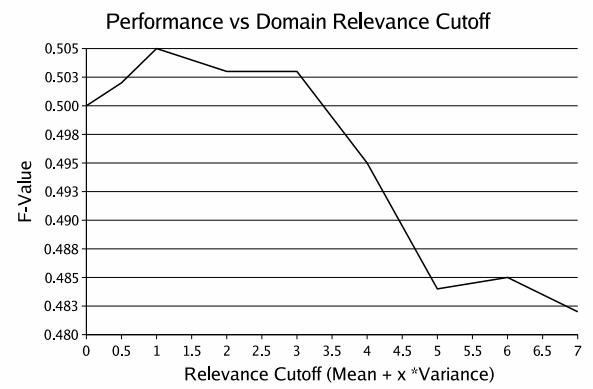

Figure 1: Performance of wsd system with respect to domain relevance cutoff

referring to the dominant domain of the context surrounding the word.

2.2 Domain relevance

(Magnini et al., 2002) introduces the notion of domain relevance. The relevance of a domain with respect to a text is represented as a positive real number. It finds that the frequency of a domain in a text does not imply its relevance in a text. They hypothesise that a domain is relevant for a text only if its frequency is significantly higher than in the texts unrelated with that domain.
In order to calculate the relation between frequency and relevance we measure the frequency

Table 2: An example of WordNet Domain entries for the senses of the word **bank**

Sense	Gloss	Domains
1	a financial institution ...	ECONOMY
2	sloping land	GEOGRAPHY, GEOLOGY
3	a supply or stock held in reserve	ECONOMY
4	a building	ARCHITECTURE, ECONOMY
5	an arrangement of similar objects	FACTOTUM
6	a container .... coin bank	ECONOMY
7	a long ridge or pile	GEOGRAPHY, GEOLOGY
8	the funds held by a gambling house	ECONOMY, PLAY
9	a slope in the turn of a road	ARCHITECTURE
10	a flight manoeuver	TRANSPORT

of each domain in the Semcor 2 corpus. Then using a normal distribution theorem it is said that if the frequency of a domain D computed on a text T is significantly higher than the mean frequency D on that corpus then D is relevant with respect to T. Our method varies from (Magnini et al., 2002) in that they calculated the domain statistics using the LOB corpus which unlike the Semcor corpus is not tagged for word sense. They included all domains of all senses of each word in the corpus in the statistical calculations where as we, due to the Semcor sense tagging, only needed to include the domains of the tagged sense of each word in the corpus. This lead to lower mean values and higher standard deviations for each domain over the corpus, and caused us to change our interpretation of *significantly higher than the mean frequency*, from exceeding the mean by two standard deviations (Magnini et al., 2002) to exceeding the mean by a single multiple of the variance. This value was found via experimentation with varying relevance cutoffs. (See Figure 1)
If a domain is not relevant in a particular text, its value is removed from the domain vector, see section 3.1. Domain relevance is only used in this manner with the real valued Wordnet Domains feature extractor, not the binary valued or selected wordnet domains feature extractors.(See Section 4)

3 WSD Infrastructure

Our word sense disambiguation system involves an extensible framework for feature extraction and feature vector construction. The framework is a 3 tiered construction:

Tier 1 - Feature Extractors This tier consists of the actual feature extractors. A feature extractor takes a context window and extracts features from it and builds a list of numerical attributes from them. A single feature extractor reserves a variable length section of the feature vector into which it inserts the values for the features it extracts. Each feature extractor extracts a class of features such as part of speech information, morphological information or domain information.

Tier 2 - Word Expert A word expert is a method that is responsible for the feature vector construction for a particular word. It organises the indivdual feature extractors input into a single feature vector. The word expert tier consists of a group of feature extractors for each word in the vocabulary.

Tier 3 - Word Expert Parliament The word expert parliament gathers parsed example data and sends examples to the word experts for processing of a particular word. The feature vectors returned from the experts are then appended with a class label (if in training stage) and the resulting feature vectors are then grouped into training or testing files and sent to the machine learner. The word expert parliament is also responsible for the construction of the disambiguation system for each target word, in that it creates the feature extractors and assigns them to a word expert according to a configuration file.

The framework is built and customised via an xml configuration file which specifies which feature extractors, and which parameters for those feature extractors, are used with each word in the vocabulary. The xml file also specifies the parameters for the machine learning. This sys-

tem can potentially be used for tagging tasks beyond word sense disambiguation e.g. POS tagging. It is essentially a feature vector construction system and by simply changing the class label on the training feature vectors, the system could be orientated to another task.

4 Features and Feature Extractors

This system has been tested with a number of feature extractors. Below is a description of the feature extractors used in this experiment:

4.1 Domain Features

The Wordnet Domains feature extractor is based on the idea that word sense is often dependant on the domain of the discourse. For example, if I am talking about computers, then the word "mouse" is more likely to refer to a computer mouse than a rodent. The domain of a discourse is established by finding the most prevalent domain amongst the word in that discourse. The domain of a particular word is found using the Wordnet Domains database (Magnini et al., 2002).

4.1.1 WordNet Domain Feature Extractor

Two versions of this feature extractor were created. The first builds a feature vector of length equal to the amount of domains at the selected depth of the WordNet Domains hierarchy and places a score in each position of this vector based upon the presence of that domain in the context window. This approach is similar to the approach taken by (Magnini et al., 2002) except that when used in our WSD infrastructure it can be combined with other features to provide a richer feature space.

4.1.2 Selected Domains Feature Extractor

The second selects the domains that will be considered based upon training data. This was done by passing over the Semcor 2.0 and Senseval2 training corpora looking for instances of the word of interest. When an occurrence of the "target" word was found the sense was noted along with the domains of the surrounding context words. This was continued until all occurrences of the word are considered. The domains selected are those that occur more often with one sense of the word than any other. These domains are considered to be indicative of the sense for that word.

In contrast to (Magnini et al., 2002) after this selection step we have an optimal set of domains from which to build our domain vector. It is hoped that this will help to reduce noise in the training.

Both of the domain information feature extractors can be set to produce binary (1 if a domain occurs, or 0 if not) and real valued (a value from 0-1 depending on significance of domain presence) vectors. Both feature extractors work on a context window of 4 sentences to the left and right of the sentence of the target word. When using the non-selected domain feature extractor domain relevance is taken into account and a dimension on the vector may be set to zero if the corresponding domain is deemed irrelevant (See section 2.2).

4.2 General Feature Extractors

For the purpose of this investigation into the a set of "General" feature extractors was constructed to provide some extra semantic and syntactic information for each word sense. These feature extractors form a basic system that has results by itself of 56% F-score for the lexical sample task, which is well above the average F Score for Senseval2 of 46.2%. The constituants of this base set of feature extractors are outlined in the next three sub-sections.

4.2.1 Context words

The context words feature extractor works in a similar way to the Selected Wordnet Domains feature extractor described above. It includes a pre-processing step in which words which are indicative of sense are identified using conditional probability. Note that this differs from the domain feature extractor as it considers the words themselves, not the domain of the word.

4.3 Collocations

The collocations feature extractor follows the basic idea of (Ng and Lee, 1996) where 9 word sequences around the target word are considered as collocations involving the word. These 9 word sequences are found by varying the left offset and right offsets of the context window around the target word. This creates word sequences of up to 4 words starting from up to 3 words to the left and ending up to 3 words to the right of the target word. See table 3.

Table 4: Results of experimentation for Senseval 2 eng-lex-sample using fine grained scoring

Feature Extractors	**Precision**	**Recall**	**F-Value**
LEMMA (baseline)	0.48	0.48	0.48
GENERAL	0.56	0.56	**0.56**
WND Binary	0.45	0.45	0.45
WND Real	0.51	0.51	**0.51**
PSWND Binary	0.47	0.46	0.47
PSWND Real	0.48	0.47	0.48
GENERAL + WND Real	0.56	0.56	**0.56**
GENERAL + PSWND Real	0.53	0.53	0.53
GENERAL + PSWND + WND Real	0.56	0.56	0.56
GENERAL + LEMMA	0.54	0.54	0.54
MAGNINI	0.67	0.25	0.36
WND Real Magnini Set	0.62	0.23	0.34

Table 3: Features for Collocations involving the word "interest" - (Ng and Lee, 1996). Left and Right indicate the position of the left and right boundaries of the context windows, with respect to the target word

Left	Right	Collocation Example
-1	-1	accrued interest
1	1	interest rate
-2	-1	principle and interest
-1	1	national interest in
1	2	interest and dividends
-3	-1	sale of an interest
-2	1	in the interest of
-1	2	an interest in a
1	3	interest on the bonds

A pre-processing step is also done with this feature extractor to identify the collocations that are indicative of sense for each word. The vector generated by this feature extractor has a position for each collocation identified which is set to 1 if the collocation exists in the example or 0 if it doesn't.

4.3.1 Other feature extractors

A number of other less sophisticated feature extractors where used to enhance the performance of the word experts. These included a part of speech identifier which considers the part of speech of the target word and context words, and a morphology feature extractor which considers the morphology of the target word and context words.

5 Method

In order to test the effect of using WordNet Domain information in supervised learning WSD a number of experiments were conducted, using different feature extractors in the framework described in section 3. Firstly the selected and the regular Wordnet domains feature extractors were tested with both binary and real valued feature vectors. Secondly the General features extractors were tested as a group with no input from the WordNet Domains feature extractors. Next, the General feature extractors were used in combination with the Wordnet Domains feature extractor, the Pre-Selected Wordnet Domains feature extractor seperatly and both together. Finally an experiment was run with only the lemma feature extractor, which simply provided the lemma of the target word as a feature. This was used in order to provide a further baseline for comparison.

5.1 Data

The data used for this research consisted of the Semcor2.0 corpus and the training and testing data from the Senseval 2 lexical sample task. Semcor is a completely word sense tagged version of sections of the Brown corpus, and thus provides a rich resource for training.

5.2 Parsing and Processing

All data was parsed using the Conexor parser. This parser tags for part of speech, morphology, as well as syntactic links. Processing involved converting the xml output of the parser to include sense information and then using our WSD infrastructure to build arff files for the Weka machine learning framework, these files consists of the feature vectors extracted by

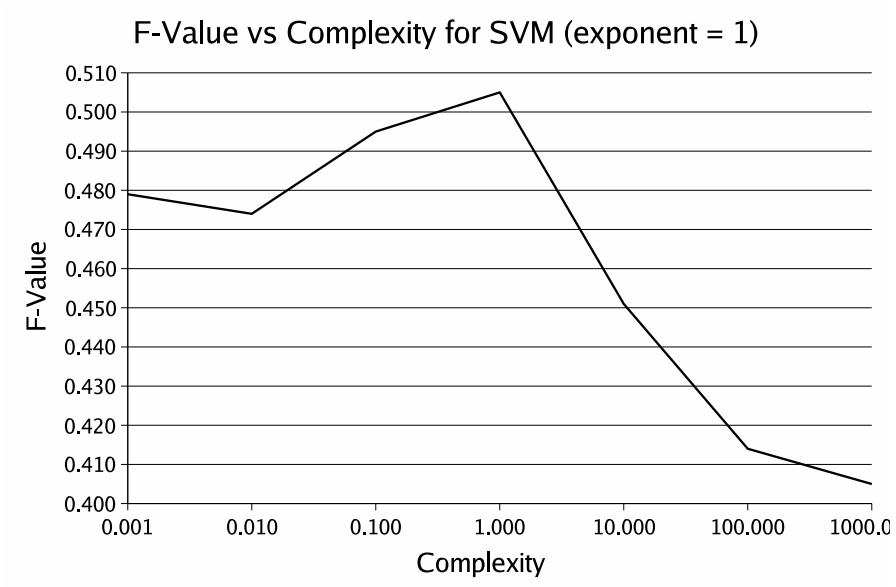

Figure 2: Performance of polynomial svm kernel with varying complexity and exponent of 1

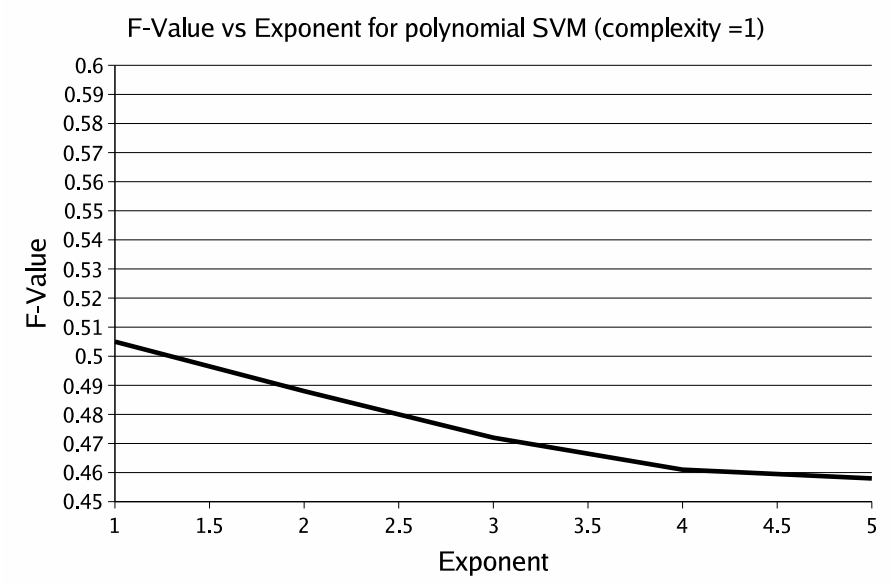

Figure 3: Performance of polynomial svm kernel with varying exponent and complexity of 1

the numerous feature extractors for each training/testing document.

5.3 Machine Learning

The Weka machine learning framework was used for this research. It involves a collection of machine learning algorithms and an infrastructure to aid in processing of data and results. Support Vector Machines were used as the machine learning algorithm and a variety of kernels and variables were tested in order to find the best parameters for the wsd task using domain information. The two kernels tested were a polynomial kernel and the RBF kernel. The polynomial kernel was tested while varying the exponent and complexity values, and the RBF kernel was tested whilst varying the gamma value. The results of these experiments are shown in figures 2,3 and 4. From this experimentation a decision was made to use a polynomial kernel with complexity 1.0 and exponent 1.0.

6 Results

In all 12 experiments were run. Each experiment consisted of building a supervised learning

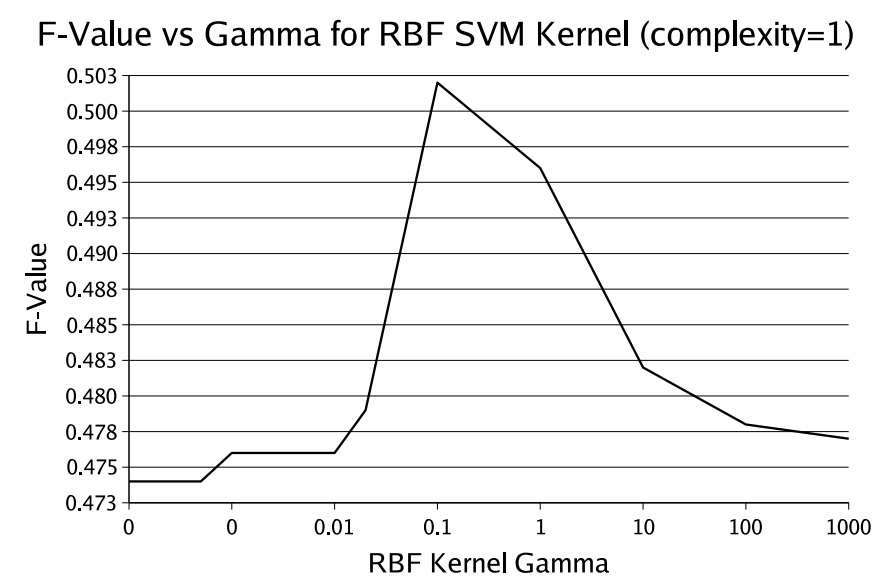

Figure 4: Performance of RBF svm kernel with varying gamma and complexity of 1

model by processing the training data (semcor and senseval) producing the feature vectors and using Weka to construct a model file. Then each model was tested using the Senseval 2 evaluation data for the english lexical sample section. The Senseval 2 scoring software was used to evaluate performance. When the PSWND and WND feature extractors were combined with the BASE feature extractors the real valued version of both these feature extractors was used as these were indicated to be better from earlier experiments. The results of these experiments are shown in table 4 which displays the precision, recall and f-value of the various feature extraction combinations for the Senseval2 lexical sample task. Table 4 also shows the results obtained by (Magnini et al., 2002) for comparison.

7 Discussion

Table 4 shows that the best way to use Wordnet domains is in a real valued non-selected fashion (WND Real). However domain information, when combined with other features has little impact (GENERAL + WND Real). This could be due to the fact that the information, relevant to word sense, provided by Wordnet Domains is already contained in the other features included in the GENERAL set, such as context words and collocations. However this argument can not be assured as it is uncertain if the same errors are being made by the GENERAL set and the Wordnet domains. The performance of the WND Real extractor is superior to that shown in (Magnini et al., 2002), f-value of 0.51 vs f-value of 0.36 and respectable in terms of Senseval 2 results (avg f-score 0.44, max f-score 0.64), however when the same experimentation is run over the subset of the Senseval data that the Magnini system attempted, the performance of our system is slightly lower (2% on f-value). This *Magnini set* is highly tuned to their method as they only make decisions on 37% of the testing corpus. This must almost certainly point to the fact that the Magnini system is a non-optimal solution across a wider data set.
The selected domains technique seems to not do well, and this is probably due to the small number of indicitive domains found during the selection process. This number could be increased by changing the conditional probability thresholds for selection. Real valued features do better than binary features, this is attributed to the richer model that a real valued feature incorporates and shows that the quantity of domain indicators in a context is important to word sense indication.
The diversity in sense determination is equally matched between WND and General features. The latter being four times larger than the former indicates a weakness in selectivity in the Wordnet domains database.

8 Conclusions

We have described a word sense disambiguation framework that can be used to easily combine

many different feature extraction techniques for the context of the word to be disambiguated. This framework was used to investigate the effect of using domain information, in particular the WordNet domain database information, on the word sense disambiguation task. This was done through a series of experiments involving two versions of a WordNet domain feature extractor and a set of General feature extractors. The performance was also measured against a simple feature extractor that took only the lemma of the target word. It was found that while WordNet domains can be used to provide a good feature for word sense disambiguation, when combined with other features such as context words and collocations the add-on effect is minimal.

9 Future Work

Further work could be done to examine the effect of using different levels of the domain hierarchy. This could change greatly the effect of WordNet domains on the WSD task. It would be interesting to see if changing the selection criteria on the Selected Domains feature extractor would improve the performance of this feature. (Magnini et al., 2002) reports better performance of WordNet domains on the Senseval all words task which has not been investigated in this research, thus it would be interesting to extend this system for the all words task. Another improvement might be made by using information such as part of speech to eliminate senses of context words in order to restrict the amount of possible domains for that word. Finally some investigation should be conducted into the errors being made by the classifier based on the GENERAL set of features and the classifier based on Wordnet domains features. This would indicate whether or not the two sets of features overlap, and if this overlap causes the combination of these two feature sets to be unsuccessful.

Acknowledgements

The word sense disambiguation architecture was jointly constructed with Ari Chanen. We would like to thank: The people at The Instituto Trentino di Cultura (ITC) for providing the WordNet Domains database; The Capital Markets CRC and The University of Sydney for financial assistance and everyone from The Sydney Language Technology Research Group for all their support.

References

Christopher J. C. Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Clara Cabezas, Philip Resnik, and Jessica Stevens. 2001. Supervised sense tagging using support vector machines.

Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press.

Philip Edmonds. 2000. Designing a task for senseval-2.

Gale, Church, and Yarowsky. 1992. One sense per discourse. In *4th ARPA Workshop on Speech and Natural Language Processing*, pages 233–237, NY. Harriman.

A. Kilgarriff and J. Rosenzweig. 2002. English senseval: Report and results.

Adam Kilgarriff. 1998. SENSEVAL: An exercise in evaluating word sense disambiguation programs. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 581–588, Granada, Spain.

A. Kilgarriff. 2000. English lexical sample task description.

Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(4):359–373.

Rada F. Mihalcea. 2002. Word sense disambiguation with pattern learning and automatic feature selection. *Natural Language Engineering*, 8(4):343–358.

Hwee Tou Ng and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In Arivind Joshi and Martha Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 40–47, San Francisco. Morgan Kaufmann Publishers.

T. Pedersen. 2001. Machine learning with lexical features: The duluth approach to senseval.

Bernhard Schölkopf and Alexander J. Smola. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press.