Supersense Tagging of Unknown Nouns in WordNet[*]

Massimiliano Ciaramita
Brown University
massi@brown.edu

Mark Johnson
Brown University
mark_johnson@brown.edu

Abstract

We present a new framework for classifying common nouns that extends named-entity classification. We used a fixed set of 26 semantic labels, which we called *supersenses*. These are the labels used by lexicographers developing WordNet. This framework has a number of practical advantages. We show how information contained in the dictionary can be used as additional training data that improves accuracy in learning new nouns. We also define a more realistic evaluation procedure than cross-validation.

1 Introduction

Lexical semantic information is useful in many natural language processing and information retrieval applications, particularly tasks that require complex inferences involving world knowledge, such as question answering or the identification of coreferential entities (Pasca and Harabagiu, 2001; Pustejovsky et al., 2002).

However, even large lexical databases such as WordNet (Fellbaum, 1998) do not include all of the words encountered in broad-coverage NLP applications. Ideally, we would like a system that automatically extends existing lexical resources by identifying the syntactic and semantic properties of unknown words. In terms of the WordNet lexical database, one would like to automatically assign unknown words a position in the synset hierarchy, introducing new synsets and extending the synset hierarchy where appropriate. Doing this accurately is a difficult problem, and in this paper we address a simpler problem: automatically determining the broad semantic class, or *supersense*, to which unknown words belong.

Systems for thesaurus extension (Hearst, 1992; Roark and Charniak, 1998), information extraction (Riloff and Jones, 1999) or named-entity recognition (Collins and Singer, 1999) each partially address this problem in different ways. The goal in these tasks is automatically tagging words with semantic labels such as “vehicle”, “organization”, “person”, etc.

In this paper we extend the named-entity recognition approach to the classification of common nouns into 26 different supersenses. Rather than define these ourselves, we adopted the 26 “lexicographer class” labels used in WordNet, which include labels such as person, location, event, quantity, etc. We believe our general approach should generalize to other definitions of supersenses.

Using the WordNet lexicographer classes as supersenses has a number of practical advantages. First, we show how information contained in the dictionary can be used as additional training data that improves the system’s accuracy. Secondly, it is possible to use a very natural evaluation procedure. A system can be trained on an earlier release of WordNet and tested on the words added in a later release,

[*] We would like to thank Thomas Hofmann, Brian Roark, and our colleagues in the Brown Laboratory for Linguistic Information Processing (BLLIP), as well as Jesse Hochstadt for his editing advice. This material is based upon work supported by the National Science Foundation under Grant No. 0085940.

1	person	7	cognition	13	attribute	19	quantity	25	plant
2	communication	8	possession	14	object	20	motive	26	relation
3	artifact	9	location	15	process	21	animal		
4	act	10	substance	16	Tops	22	body		
5	group	11	state	17	phenomenon	23	feeling		
6	food	12	time	18	event	24	shape		

Table 1. Lexicographer class labels, or *supersenses*.

since these labels are constant across different releases. This new evaluation defines a realistic lexical acquisition task which is well defined, well motivated and easily standardizable.

The heart of our system is a multiclass perceptron classifier (Crammer and Singer, 2002). The features used are the standard ones used in word-sense classification and named-entity extraction tasks, i.e., collocation, spelling and syntactic context features.

The experiments presented below show that when the classifier also uses the data contained in the dictionary its accuracy improves over that of a traditionally trained classifier. Finally, we show that there are both similarities and differences in the results obtained with the new evaluation and standard cross-validation. This might suggest that in fact that the new evaluation defines a more realistic task.

The paper is organized as follows. In Section 2 we discuss the problem of unknown words and the task of semantic classification. In Section 3 we describe the WordNet lexicographer classes, how to extract training data from WordNet, the new evaluation method and the relation of this task to named-entity classification. In Section 4 we describe the experimental setup, and in Section 5 we explain the averaged perceptron classifier used. In Section 6 and 7 we discuss the results and the two evaluations.

2 Unknown Words and Semantic Classification

Language processing systems make use of "dictionaries", i.e., lists that associate words with useful information such as the word's frequency or syntactic category. In tasks that also involve inferences about world knowledge, it is useful to know something about the meaning of the word. This lexical semantic information is often modeled on what is found in normal dictionaries, e.g., that "irises" are flowers or that "exane" is a solvent.

This information can be crucial in tasks such as question answering - e.g., to answer a question such as "What kind of *flowers* did Van Gogh paint?" (Pasca and Harabagiu, 2001) - or the individuation of co-referential expressions, as in the passage "... the prerun can be performed with $exane_i$ ... this $solvent_i$ can be considered ..." (Pustejovsky et al., 2002).

Lexical semantic information can be extracted from existing dictionaries such as WordNet. However, these resources are incomplete and systems that rely on them often encounter unknown words, even if the dictionary is large. As an example, in the Bllip corpus (a very large corpus of Wall Street Journal text) the relative frequency of common nouns that are unknown to WordNet 1.6 is approximately 0.0054; an unknown noun occurs, on average, every eight sentences. WordNet 1.6 lists 95,000 noun types. For this reason the importance of issues such as automatically building, extending or customizing lexical resources has been recognized for some time in computational linguistics (Zernik, 1991).

Solutions to this problem were first proposed in AI in the context of story understanding, cf. (Granger, 1977). The goal is to label words using a set of semantic labels specified by the dictionary. Several studies have addressed the problem of expanding one semantic category at a time, such as "vehicle" or "organization", that are relevant to a particular task (Hearst, 1992; Roark and Charniak, 1998; Riloff and Jones, 1999). In named-entity classification a large set of named entities (proper nouns) are classified using a comprehensive set of semantic labels such as "organization", "person", "location" or "other" (Collins and Singer, 1999). This latter approach assigns all named entities in the data set a semantic label. We extend this approach to the classification of common nouns using a suitable set of semantic classes.

3 Lexicographer Classes for Noun Classification

3.1 WordNet Lexicographer Labels

WordNet (Fellbaum, 1998) is a broad-coverage machine-readable dictionary. Release 1.71 of the English version lists about 150,000 entries for all open-class words, mostly nouns (109,000 types), but also verbs, adjectives, and adverbs. WordNet is organized as a network of lexicalized concepts, sets of synonyms called *synsets*; e.g., the nouns {chairman, chairwoman, chair, chairperson} form a synset. A word that belongs to several synsets is *ambiguous*.

To facilitate the development of WordNet, lexicographers organize synsets into several domains, based on syntactic category and semantic coherence. Each noun synset is assigned one out of 26 broad categories[1]. Since these broad categories group together very many synsets, i.e., word senses, we call them *supersenses*. The supersense labels that WordNet lexicographers use to organize nouns are listed in Table 1[2]. Notice that since the lexicographer labels are assigned to synsets, often ambiguity is preserved even at this level. For example, *chair* has three supersenses: “person”, “artifact”, and “act”.

This set of labels has a number of attractive features for the purposes of lexical acquisition. It is fairly general and therefore small. The reasonable size of the label set makes it possible to apply state-of-the-art machine learning methods. Otherwise, classifying new words at the synset level defines a multiclass problem with a huge class space - more than 66,000 noun synsets in WordNet 1.6, more than 75,000 in the newest release, 1.71 (cf. also (Ciaramita, 2002) on this problem). At the same time the labels are not too abstract or vague. Most of the classes seem natural and easily recognizable. That is probably why they were chosen by the lexicographers to facilitate their task. But there are more important practical and methodological advantages.

3.2 Extra Training Data from WordNet

WordNet contains a great deal of information about words and word senses.The information contained in the dictionary’s glosses is very similar to what is typically listed in normal dictionaries: synonyms, definitions and example sentences. This suggests a very simple way in which it can be put into use: it can be compiled into training data for supersense labels. This data can then be added to the data extracted from the training corpus.

For several thousand concepts WordNet’s glosses are very informative. The synset “chair” for example looks as follows:

- $chair$: president, chairman, chairwoman, chair, chairperson – (the officer who presides at the meetings of an organization); “address your remarks to the chairperson”.

In WordNet 1.6, 66,841 synsets contain definitions (in parentheses above), and 6,147 synsets contain example sentences (in quotation marks). As we show below, this information about word senses is useful for supersense tagging. Presumably this is because if it can be said of a “chairperson” that she can “preside at meetings” or that “a remark” can be “addressed to her”, then logically speaking these things can be said of the superordinates of “chairperson”, like “person”, as well.

Therefore information at the synset level is relevant also at the supersense level. Furthermore, while individually each gloss doesn’t say too much about the narrow concept it is attached to (at least from a machine learning perspective) at the supersense level this information accumulates. In fact it forms a small corpus of supersense-annotated data that can be used to train a classifier for supersense tagging of words or for other semantic classification tasks.

3.3 Evaluation Methods

Formulating the problem in this fashion makes it possible to define also a very natural evaluation procedure. Systems can be trained on nouns listed in a given release of WordNet and tested on the nouns introduced in a later version. The set of lexicographer labels remains constant and can be used across different versions.

In this way systems can be tested on a more realistic lexical acquisition task - the same task that lexicographers carried out to extend the database. The task is then well defined and motivated, and easily standardizable.

[1] There are also 15 lexicographer classes for verbs, 3 for adjectives and 1 for adverbs.

[2] The label “Tops” refers to about 40 very general synsets, such as “phenomenon” “entity” “object” etc.

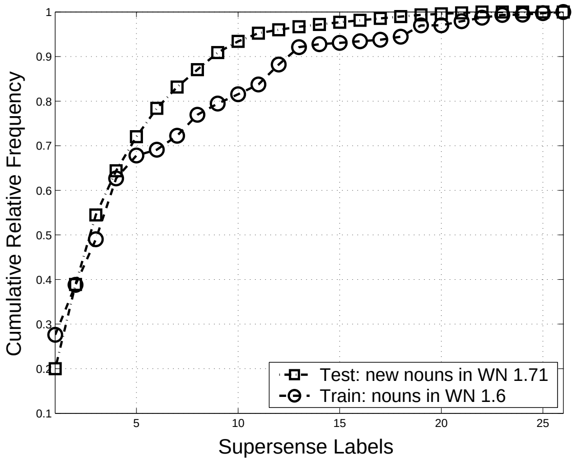

Figure 1. Cumulative distribution of supersense labels in Bllip.

3.4 Relation to Named-Entity Tasks

The categories typically used in named-entity recognition tasks are a subset of the noun supersense labels: “person”, “location”, and “group”. Small label sets like these can be sufficient in named-entity recognition. Collins and Singer (1999) for example report that 88% of the named entities occurring in their data set belong to these three categories (Collins and Singer, 1999).

The distribution of common nouns, however, is more uniform. We estimated this distribution by counting the occurrences of 744 unambiguous common nouns newly introduced in WordNet 1.71. Figure 1 plots the cumulative frequency distribution of supersense tokens; the labels are ordered by decreasing relative frequency as in Table 1.

The most frequent supersenses are “person”, “communication”, “artifact” etc. The three most frequent supersenses account for a little more of 50% of all tokens, and 9 supersenses account for 90% of all tokens. A larger number of labels is needed for supersense tagging than for named-entity recognition. The figure also shows the distribution of labels for all unambiguous tokens in WordNet 1.6; the two distributions are quite similar.

4 Experiments

The “new” nouns in WordNet 1.71 and the “old” ones in WordNet 1.6 constitute the test and training data that we used in our word classification experiments. Here we describe the experimental setup: training and test data, and features used.

4.1 Training data

We extracted from the Bllip corpus all occurrences of nouns that have an entry in WordNet 1.6. Bllip (BLLIP, 2000) is a 40-million-word syntactically parsed corpus. We used the parses to extract the syntactic features described below. We then removed all ambiguous nouns, i.e., nouns that are tagged with more than one supersense label (72% of the tokens, 28.9% of the types). In this way we avoided dealing with the problem of ambiguity[3].

We extracted a feature vector for each noun instance using the feature set described below. Each vector is a training instance. In addition we compiled another training set from the example sentences and from the definitions in the noun database of WordNet 1.6. Overall this procedure produced 787,186 training instances from Bllip, 66,841 training instances from WordNet’s definitions, and 6,147 training instances from the example sentences.

4.2 Features

We used a mix of standard features used in word sense disambiguation, named-entity classification and lexical acquisition. The following sentence illustrates them: “The *art-students*, nine teen-agers, read the book”, *art-students* is the tagged noun:

1. part of speech of the neighboring words: $\texttt{P}_{-1} = \texttt{DT}$, $\texttt{P}_0 = \texttt{NNS}$, $\texttt{P}_{+1} = \texttt{CD}$, ...
2. single words in the surrounding context: $\texttt{C} = \texttt{read}$, $\texttt{C} = \texttt{book}$, $\texttt{C} = \texttt{class}$, $\texttt{C} = \texttt{the}$, ...
3. bigrams and trigrams: $\texttt{C}_{-1,+1} = \texttt{the\_nine}$, $\texttt{C}_{-1,-1} = \texttt{the}$, $\texttt{C}_{+1,+2} = \texttt{nine\_teen} - \texttt{agers}$, ...
4. syntactically governed elements under a given phrase: $\texttt{G}_1 = \texttt{the\_NP}$
5. syntactically governing elements under a given phrase: $\texttt{G}_2 = \texttt{read\_S}$
6. coordinates/appositives: $\texttt{CO} = \texttt{teen} - \texttt{agers}$
7. spelling/morphological features: prefixes, suffixes, complex morphology: $\texttt{MP} = \texttt{a}$, $\texttt{MP} = \texttt{ar}$ ... $\texttt{MS} = \texttt{s}$, $\texttt{MS} = \texttt{ts}$ ... $\texttt{MC} = \texttt{art}$, $\texttt{MC} = \texttt{student}$ ...

[3] A simple option to deal with ambiguous words would be to distribute an ambiguous noun’s counts to all its senses. However, in preliminary experiments we found that a better accuracy is achieved using only non-ambiguous nouns. We will investigate this issue in future research.

Open class words were morphologically simplified with the "morph" function included in WordNet. We parsed the WordNet definitions and example sentences with the same syntactic parser used for Bllip (Charniak, 2000).

It is not always possible to identify the noun that represents the synset in the WordNet glosses. For example, in the gloss for the synset *relegation* the example sentence is "He has been relegated to a post in Siberia", where a verb is used instead of the noun. When it was possible to identify the target noun the complete feature set was used; otherwise only the surrounding-word features (2) and the spelling features (7) of all synonyms were used. With the definitions it is much harder to individuate the target; consider the definition "a member of the genus Canis" for *dog*. For all definitions we used only the reduced feature set. One training instance per synset was extracted from the example sentences and one training instance from the definitions. Overall, in the experiments we performed we used around 1.5 million features.

4.3 Evaluation

In a similar way to how we produced the training data we compiled a test set from the Bllip corpus. We found all instances of nouns that are not in WordNet 1.6 but are listed in WordNet 1.71 with only one supersense. The majority of the novel nouns in WordNet 1.71 are unambiguous (more than 90%). There were 744 new noun types, with a total frequency of 9,537 occurrences. We refer to this test set as Test$_{1.71}$.

We also randomly removed 755 noun types (20,394 tokens) from the training data and used them as an alternative test set. We refer to this other test set as Test$_{1.6}$. We then ran experiments using the averaged multiclass perceptron.

5 The Multiclass Averaged Perceptron

We used a multiclass averaged perceptron classifier, which is an "ultraconservative" on-line learning algorithm (Crammer and Singer, 2002), that is a multiclass extension of the standard perceptron learning to the multiclass case. It takes as input a training set $S = (x_i, y_i)_{i=1}^n$, where each instance $x_i \in \mathbb{R}^d$ represents an instance of a noun and $y_j \in Y$. Here Y is the set of supersenses defined by WordNet. Since for training and testing we used only unambiguous words there is always exactly one label per instance. Thus S summarizes n word tokens that belong to the dictionary, where each instance i is represented as a vector of features x_i extracted from the context in which the noun occurred; d is the total number of features; and y_i is the true label of x_i.

Algorithm 1 Multiclass Perceptron

```
input training data $(x_i, y_i)_{i=1}^n$, $\mathbf{V} = 0$
repeat
  for $i = 1, ..., n$ do
    if $H(x_i; \mathbf{V}) \neq y_i$ then
      $v_{y_i} \leftarrow v_{y_i} + x_i$
      $E_i = \{y \in Y : \langle v_y, x_i \rangle > \langle v_{y_i}, x_i \rangle\}$
      for $y \in E_i$ do
        $v_y \leftarrow v_y - \frac{1}{|E_i|} x_i$
      end for
    end if
  end for
until no more mistakes
```

In general, a multiclass classifier for the dictionary is a function $H : \mathbb{R}^n \rightarrow Y$ that maps feature vectors x to one of the possible supersenses of WordNet. In the multiclass perceptron, one introduces a weight vector $v_y \in \mathbb{R}^d$ for every $y \in Y$ and defines H implicitly by the so-called winner-take-all rule

$$H(x; \mathbf{V}) = \arg\max_{y \in Y} \langle v_y, x \rangle \,. \tag{1}$$

Here $\mathbf{V} \in \mathbb{R}^{k \times d}$ refers to the matrix of weights, with every column corresponding to one of the weight vectors v_y.

The learning algorithm works as follows: Training patterns are presented one at a time in the standard on-line learning setting. Whenever $H(x_i; \mathbf{V}) \neq y_i$ an update step is performed; otherwise the weight vectors remain unchanged. To perform the update, one first computes the error set E_i containing those class labels that have received a higher score than the correct class:

$$E_i = \{y \in Y : \langle v_y, x_i \rangle > \langle v_{y_i}, x_i \rangle\} \tag{2}$$

An ultraconservative update scheme in its most general form is then defined as follows: Update $v_y \leftarrow$

$v_y + \tau_y x_i$ with learning rates fulfilling the constraints $\tau_{y_i} = 1$, $\sum_{y \neq y_i} \tau_y = -1$, and $\tau_y = 0$ for $y \notin E_i \cup \{y_i\}$. Hence changes are limited to v_y for $y \in E_i \cup \{y_i\}$. The sum constraint ensures that the update is balanced, which is crucial to guaranteeing the convergence of the learning procedure (cf. (Crammer and Singer, 2002)). We have focused on the simplest case of uniform update weights, $\tau_y = -\frac{1}{|E_i|}$ for $y \in E_i$. The algorithm is summarized in Algorithm 1.

Notice that the multiclass perceptron algorithm learns all weight vectors in a coupled manner, in contrast to methods that perform multiclass classification by combining binary classifiers, for example, training a classifier for each class in a one-against-the-rest manner.

The averaged version of the perceptron (Collins, 2002), like the voted perceptron (Freund and Schapire, 1999), reduces the effect of over-training. In addition to the matrix of weight vectors $\mathbf{V}$ the model keeps track for each feature f of each value it assumed during training, f_j, and the number of consecutive training instance presentations during which this weight was not changed, or “life span”, $ls(f_j)$. When training is done these weights are averaged and the final averaged weight f_{avg} of feature f is computed as

$$f_{avg} = \frac{\sum_j f_j ls(f_j)}{\sum_j ls(f_j)} \qquad (3)$$

For example, if there is a feature weight that is not updated until example 500, at which point it is incremented to value 1, and is not touched again until after example 1000, then the average weight of that feature in the averaged perceptron at example 750 will be: $\frac{(0*500+1*250)}{(500+250)}$, or 1/3. At example 1000 it will be 1/2, etc. We used the averaged model for evaluation and parameter setting; see below. Figure 2 plots the results on test data of both models. The average model produces a better-performing and smoother output.

5.1 Parameters Setting

We used an implementation with full, i.e., not sparse, representation of the matrix for the perceptron. Training and test are fast, at the expense of a slightly greater memory load. Given the great number of features, we couldn’t use the full training set from the Bllip corpus. Instead we randomly sampled from roughly half of the available training data, yielding around 400,000 instances, the size of the training is close to 500,000 instances with also the WordNet data. When training to test on Test$_{1,6}$, we removed from the WordNet training set the synsets relative to the nouns in Test$_{1.6}$.

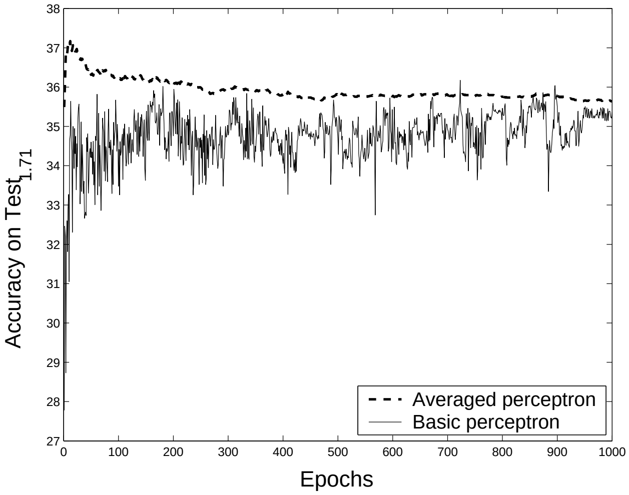

Figure 2. Results on test of the normal and averaged perceptron

The only adjustable parameter to set is the number of passes on the training data, or *epochs*. While testing on Test$_{1.71}$ we set this parameter using Test$_{1.6}$, and vice versa for Test$_{1.6}$. The estimated values for the stopping iterations were very close at roughly ten passes. As Figure 2 shows, the great amount of data requires many passes over the data, around 1,000, before reaching convergence (on Test$_{1.71}$).

6 Results

The classifier outputs the estimated supersense label of each *instance* of each unknown noun type. The label $L(n)$ of a noun type n is obtained by voting[4]:

$$L(n) = \arg\max_{y \in Y} \sum_{x \in n} [\![H(x; \mathbf{V}) = y]\!] \qquad (4)$$

where $[\![.]\!]$ is the indicator function and $x \in n$ means that x is a token of type n. The score on n is 1 if

[4] During preliminary experiments we tried also creating one single aggregate pattern for each test noun type but this method produced worse results.

Method	Token	Type	Test set
Baseline	20.0	27.8	$\text{Test}_{1.71}$
AP-B-55	35.9	50.7	
AP-B-65	36.1	50.8	
AP-B-55+WN	36.9	52.9	
Baseline	24.1	21.0	$\text{Test}_{1.6}$
AP-B-55	47.4	47.7	
AP-B-65	47.9	48.3	
AP-B-55+WN	52.3	53.4	

Table 2. Experimental results.

$L(n) = Y(n)$, where $Y(n)$ is the correct label for n, and 0 otherwise.

Table 2 summarizes the results of the experiments on $\text{Test}_{1.71}$ (upper half) and on $\text{Test}_{1.6}$ (bottom half). A baseline was computed that always selected the most frequent label in the training data, "person", which is also the most frequent in both $\text{Test}_{1.6}$ and $\text{Test}_{1.71}$. The baseline performances are in the low twenties. The first and second columns report performance on tokens and types respectively.

The classifiers' results are averages over 50 trials in which a fraction of the Bllip data was randomly selected. One classifier was trained on 55% of the Bllip data (AP-B-55). An identical one was trained on the same data and, additionally, on the WordNet data (AP-B-55+WN). We also trained a classifier on 65% of the Bliip data (AP-B-65). Adding the WordNet data to this training set was not possible because of memory limitations. The model also trained on WordNet outperforms on both test sets those trained only on the Bllip data. A paired t-test proved the difference between models with and without WordNet data to be statistically significant. The "least" significant difference is between AP-B-65 and AP-B-55+WN (token) on $\text{Test}_{1.6}$: $\alpha = 0.003$. In all other cases the α-level is much smaller.

These results seem to show that the positive impact of the WordNet data is not simply due to the fact that there is more training data[5]. Adding the WordNet data seems more effective than adding an equivalent amount of standard training data. Figure 3 plots the results of the last set of (single trial) experiments we performed, in which we varied the amount of Bllip data to be added to the WordNet one. The model with WordNet data often performs better than the model trained only on Bllip data even when the latter training set is much larger.

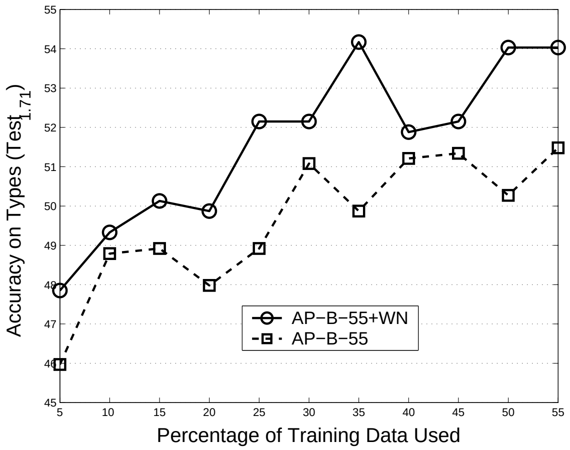

Figure 3. Results on $\text{Test}_{1.71}$ incrementing the amount of training data.

Two important reasons why the WordNet data is particularly good are, in our opinion, the following. The data is less noisy because it is extracted from sentences and definitions that are always "pertinent" to the class label. The data also contains instances of disambiguated polysemous nouns, which instead were excluded from the Bllip training. This means that disambiguating the training data is important; unfortunately this is not a trivial task. Using the WordNet data provides a simple way of getting at least some information from ambiguous nouns.

7 Differences Between Test Sets

The type scores on both evaluations produced similar results. This finding supports the hypothesis that the two evaluations are similar in difficulty, and that the two versions of WordNet are not inconsistent in the way they assign supersenses to nouns.

The evaluations show, however, very different patterns at the token level. This might be due to the fact that the label distribution of the training data is more similar to $\text{Test}_{1.6}$ than to $\text{Test}_{1.71}$. In particular, there are many new nouns in $\text{Test}_{1.71}$ that belong to "abstract" classes[6], which seem harder to learn. Abstract classes are also more confusable; i.e., mem-

[5] Notice that 10% of the Bllip data is approximately the size of the WordNet data and therefore AP-B-65 and AP-B-55+WN are trained on roughly the same amount of data.

[6] Such as "communication" (e.g., reaffirmation) or "cognition" (e.g., mind set).

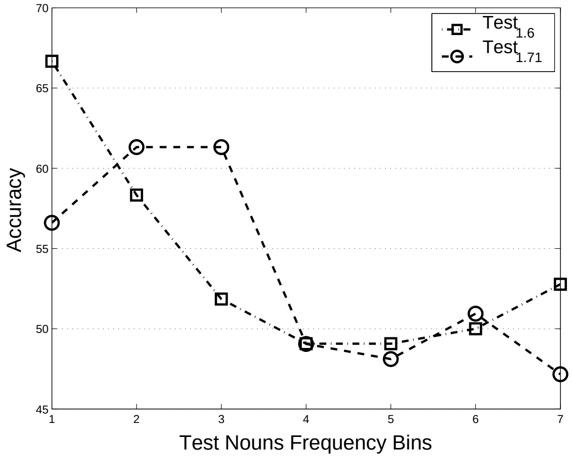

Figure 4. Results on types for Test$_{1.6}$ and Test$_{1.71}$ ranked by the frequency of the test words.

bers of these classes are frequently mis-classified with the same wrong label. A few very frequently mis-classified pairs are communication/act, communication/person and communication/artifact.

As a result of the fact that abstract nouns are more frequent in Test$_{1.71}$ than in Test$_{1.6}$ the accuracy on tokens is much worse in the new evaluation than in the more standard one. This has an impact also on the type scores. Figure 4 plots the results on types for Test$_{1.6}$ and Test$_{1.71}$ grouped in bins of test noun types ranked by decreasing frequency. It shows that the first bin is harder in Test$_{1.71}$ than in Test$_{1.6}$.

Overall, then, it seems that there are similarities but also important differences between the evaluations. Therefore the new evaluation might define a more realistic task than cross-validation.

8 Conclusion

We presented a new framework for word sense classification, based on the WordNet lexicographer classes, that extends named-entity classification. Within this framework it is possible to use the information contained in WordNet to improve classification and define a more realistic evaluation than standard cross-validation. Directions for future research include the following topics: disambiguation of the training data, e.g. during training as in co-training; learning unknown ambiguous nouns, e.g., studying the distribution of the labels the classifier guessed for the individual tokens of the new word.

References

BLLIP. 2000. *1987-1989 WSJ Corpus Release 1*. Linguistic Data Consortium.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*.

M. Ciaramita. 2002. Boosting Automatic Lexical Acquisition with Morphological Information. In *Proceedings of the Workshop on Unsupervised Lexical Acquisition, ACL-02*.

M. Collins and Y. Singer. 1999. Unsupervised Models for Named Entity Classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

M. Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-02)*, pages 1–8.

K. Crammer and Y. Singer. 2002. Ultraconservative Online Algorithms for Multiclass Problems. *Technical Report [2001-18], School of Computer Science and Engineering, Hebrew University, Jerusalem, Israel*.

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Y. Freund and R. Schapire. 1999. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, 37.

R. Granger. 1977. FOUL-UP: A Program that Figures Out Meanings of Words from Context. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*.

M. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*.

M. Pasca and S.H. Harabagiu. 2001. The Informative Role of WordNet in Open-Domain Question Answering. In *NAACL 2001 Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*.

J. Pustejovsky, A. Rumshisky, and J. Castaño. 2002. Rerendering Semantic Ontologies: Automatic Extensions to UMLS through Corpus Analytics. In *In Proceedings of REC 2002 Workshop on Ontologies and Lexical Knowledge Bases*.

E. Riloff and R. Jones. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*.

B. Roark and E. Charniak. 1998. Noun-Phrase Co-Occurrence Statistics for Semi-Automatic Semantic Lexicon Construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*.

U. Zernik. 1991. Introduction. In U. Zernik, editor, *Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon*. Lawrence Erlbaum Associates.