Unsupervised methods for developing taxonomies by combining syntactic and statistical information

Appeared in Human Langauge Technology / Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL), Edmonton, Canada, May 2003, pages 276–283

Dominic Widdows
Center for the Study of Language and Information, Stanford University
dwiddows@csli.stanford.edu

Abstract

This paper describes an unsupervised algorithm for placing unknown words into a taxonomy and evaluates its accuracy on a large and varied sample of words. The algorithm works by first using a large corpus to find semantic neighbors of the unknown word, which we accomplish by combining latent semantic analysis with part-of-speech information. We then place the unknown word in the part of the taxonomy where these neighbors are most concentrated, using a class-labelling algorithm developed especially for this task. This method is used to reconstruct parts of the existing WordNet database, obtaining results for common nouns, proper nouns and verbs. We evaluate the contribution made by part-of-speech tagging and show that automatic filtering using the class-labelling algorithm gives a fourfold improvement in accuracy.

1 Introduction

The importance of automatic methods for enriching lexicons, taxonomies and knowledge bases from free text is well-recognized. For rapidly changing domains such as current affairs, static knowledge bases are inadequate for responding to new developments, and the cost of building and maintaining resources by hand is prohibitive.

This paper describes experiments which develop automatic methods for taking an original taxonomy as a skeleton and fleshing it out with new terms which are discovered in free text. The method is completely automatic and it is completely unsupervised apart from using the original taxonomic skeleton to suggest possible classifications for new terms. We evaluate how accurately our methods can reconstruct the WordNet taxonomy (Fellbaum, 1998).

The problem of enriching the lexical information in a taxonomy can be posed in two complementary ways. Firstly, given a particular taxonomic class (such as *fruit*) one could seek members of this class (such as *apple, banana*). This problem is addressed by Riloff and Shepherd (1997), Roark and Charniak (1998) and more recently by Widdows and Dorow (2002). Secondly, given a particular word (such as *apple*), one could seek suitable taxonomic classes for describing this object (such as *fruit, foodstuff*). The work in this paper addresses the second of these questions.

The goal of automatically placing new words into a taxonomy has been attempted in various ways for at least ten years (Hearst and Schütze, 1993). The process for placing a word w in a taxonomy T using a corpus C often contains some version of the following stages:

- For a word w, find words from the corpus C whose occurrences are similar to those of w. Consider these the 'corpus-derived neighbors' $N(w)$ of w.
- Assuming that at least some of these neighbors are already in the taxonomy T, map w to the place in the taxonomy where these neighbors are most concentrated.

Hearst and Schütze (1993) added 27 words to WordNet using a version of this process, with a 63% accuracy at assigning new words to one of a number of disjoint WordNet 'classes' produced by a previous algorithm. (Direct comparison with this result is problematic since the number of classes used is not stated.) A more recent example is the top-down algorithm of Alfonseca and Manandhar (2001), which seeks the node in T which shares the most collocational properties with the word w, adding 42 concepts taken from *The Lord of the Rings* with an accuracy of 28%.

The algorithm as presented above leaves many degrees of freedom and open questions. What methods should be used to obtain the corpus-derived neighbors $N(w)$? This question is addressed in Section 2. Given a collection of neighbors, how should we define a "place in the taxonomy where these neighbors are most concentrated?" This question is addressed in Section 3, which

defines a robust class-labelling algorithm for mapping a list of words into a taxonomy. In Section 4 we describe experiments, determining the accuracy with which these methods can be used to reconstruct the WordNet taxonomy. To our knowledge, this is the first such evaluation for a large sample of words. Section 5 discusses related work and other problems to which these techniques can be adapted.

2 Finding semantic neighbors: Combining latent semantic analysis with part-of-speech information.

There are many empirical techniques for recognizing when words are similar in meaning, rooted in the idea that "you shall know a word by the company it keeps" (Firth, 1957). It is certainly the case that words which repeatedly occur with similar companions often have related meanings, and common features used for determining this similarity include shared collocations (Lin, 1999), co-occurrence in lists of objects (Widdows and Dorow, 2002) and latent semantic analysis (Landauer and Dumais, 1997; Hearst and Schütze, 1993).

The method used to obtain semantic neighbors in our experiments was a version of latent semantic analysis, descended from that used by Hearst and Schütze (1993, §4). First, 1000 frequent words were chosen as column labels (after removing stopwords (Baeza-Yates and Ribiero-Neto, 1999, p. 167)). Other words were assigned co-ordinates determined by the number of times they occured within the same context-window (15 words) as one of the 1000 column-label words in a large corpus. This gave a matrix where every word is represented by a row-vector determined by its co-occurence with frequently occuring, meaningful words. Since this matrix was very sparse, singular value decomposition (known in this context as *latent semantic analysis* (Landauer and Dumais, 1997)) was used to reduce the number of dimensions from 1000 to 100. This reduced vector space is called WordSpace (Hearst and Schütze, 1993, §4). Similarity between words was then computed using the cosine similarity measure (Baeza-Yates and Ribiero-Neto, 1999, p. 28). Such techniques for measuring similarity between words have been shown to capture semantic properties: for example, they have been used successfully for recognizing synonymy (Landauer and Dumais, 1997) and for finding correct translations of individual terms (Widdows et al., 2002).

The corpus used for these experiments was the British National Corpus, which is tagged for parts-of-speech. This enabled us to build syntactic distinctions into WordSpace — instead of just giving a vector for the string *test* we were able to build separate vectors for the nouns, verbs and adjectives *test*. An example of the contribution of part-of-speech information to extracting semantic neighbors of the word *fire* is shown in Table 2. As can be seen, the noun *fire* (as in the substance/element) and the verb *fire* (mainly used to mean firing some sort of weapon) are related to quite different areas of meaning. Building a single vector for the string *fire* confuses this distinction — the neighbors of *fire* treated just as a string include words related to both the meaning of *fire* as a noun (more frequent in the BNC) and as a verb.

Part of the goal of our experiments was to investigate the contribution that this part-of-speech information made for mapping words into taxonomies. As far as we are aware, these experiments are the first to investigate the combination of latent semantic indexing with part-of-speech information.

3 Finding class-labels: Mapping collections of words into a taxonomy

Given a collection of words or multiword expressions which are semantically related, it is often important to know what these words have in common. All adults with normal language competence and world knowledge are adept at this task — we know that *plant, animal* and *fungus* are all living things, and that *plant, factory* and *works* are all kinds of buildings. This ability to classify objects, and to work out which of the possible classifications of a given object is appropriate in a particular context, is essential for understanding and reasoning about linguistic meaning. We will refer to this process as *class-labelling*.

The approach demonstrated here uses a hand-built taxonomy to assign class-labels to a collection of similar nouns. As with much work of this nature, the taxonomy used is WordNet (version 1.6), a freely-available broad-coverage lexical database for English (Fellbaum, 1998). Our algorithm finds the hypernyms which subsume as many as possible of the original nouns, as closely as possible [1]. The concept v is said to be a *hypernym* of w if w is a kind of v. For this reason this sort of a taxonomy is sometimes referred to as an 'IS_A hierarchy'. For example, the possible hypernyms given for the word *oak* in WordNet 1.6 are

> oak $\Rightarrow$ wood $\Rightarrow$ plant material $\Rightarrow$ material, stuff $\Rightarrow$ substance, matter $\Rightarrow$ object, physical object $\Rightarrow$ entity, something

[1] Another method which could be used for class-labelling is given by the conceptual density algorithm of Agirre and Rigau (1996), which those authors applied to word-sense disambiguation. A different but related idea is presented by Li and Abe (1998), who use a principle from information theory to model selectional preferences for verbs using different classes from a taxonomy. Their algorithm and goals are different from ours: we are looking for a single class-label for semantically related words, whereas for modelling selectional preferences several classes may be appropriate.

fire (string only)		fire_nn1		fire_vvi	
fire	1.000000	fire_nn1	1.000000	fire_vvi	1.000000
flames	0.709939	flames_nn2	0.700575	guns_nn2	0.663820
smoke	0.680601	smoke_nn1	0.696028	firing_vvg	0.537778
blaze	0.668504	brigade_nn1	0.589625	cannon_nn0	0.523442
firemen	0.627065	fires_nn2	0.584643	gun_nn1	0.484106
fires	0.617494	firemen_nn2	0.567170	fired_vvd	0.478572
explosion	0.572138	explosion_nn1	0.551594	detectors_nn2	0.477025
burning	0.559897	destroyed_vvn	0.547631	artillery_nn1	0.469173
destroyed	0.558699	burning_aj0	0.533586	attack_vvb	0.468767
brigade	0.532248	blaze_nn1	0.529126	firing_nn1	0.459000
arson	0.528909	arson_nn1	0.522844	volley_nn1	0.458717
accidental	0.519310	alarms_nn2	0.512332	trained_vvn	0.447797
chimney	0.489577	destroyed_vvd	0.512130	enemy_nn1	0.445523
blast	0.488617	burning_vvg	0.502052	alert_aj0	0.443610
guns	0.487226	burnt_vvn	0.500864	shoot_vvi	0.443308
damaged	0.484897	blast_nn1	0.498635	defenders_nn2	0.438886

Table 1: Semantic neighbors of *fire* with different parts-of-speech. The scores are cosine similarities

> oak, oak tree $\Rightarrow$ tree $\Rightarrow$ woody plant, ligneous plant $\Rightarrow$ vascular plant, tracheophyte $\Rightarrow$ plant, flora, plant life $\Rightarrow$ life form, organism, being, living thing $\Rightarrow$ entity, something

Let S be a set of nouns or verbs. If the word $w \in S$ is recognized by WordNet, the WordNet taxonomy assigns to w an ordered set of hypernyms $H(w)$.

Consider the union

$$\mathcal{H} = \bigcup_{w \in S} H(w).$$

This is the set of all hypernyms of any member of S. Our intuition is that the most appropriate class-label for the set S is the hypernym $h \in \mathcal{H}$ which subsumes as many as possible of the members of S as closely as possible in the hierarchy. There is a trade-off here between subsuming 'as many as possible' of the members of S, and subsuming them 'as closely as possible'. This line of reasoning can be used to define a whole collection of 'class-labelling algorithms'.

For each $w \in S$ and for each $h \in \mathcal{H}$, define the *affinity score function* $\alpha(w, h)$ between w and h to be

$$\alpha(w,h) = \begin{cases} f(\mathrm{dist}(w,h)) & \text{if } h \in H(w) \\ -g(w,h) & \text{if } h \notin H(w), \end{cases} \qquad (1)$$

where $\mathrm{dist}(w, h)$ is a measure of the distance between w and h, f is some positive, monotonically decreasing function, and g is some positive (possibly constant) function.

The function f accords 'positive points' to h if h subsumes w, and the condition that f be monotonically decreasing ensures that h gets more positive points the closer it is to w. The function g subtracts 'penalty points' if h does not subsume w. This function could depend in many ways on w and h — for example, there could be a smaller penalty if h is a very specific concept than if h is a very general concept.

The distance measure $\mathrm{dist}(w, h)$ could take many forms, and there are already a number of distance measures available to use with WordNet (Budanitsky and Hirst, 2001). The easiest method for assigning a distance between words and their hypernyms is to count the number of intervening levels in the taxonomy. This assumes that the distance in specificity between ontological levels is constant, which is of course not the case, a problem addressed by Resnik (1999).

Given an appropriate affinity score, it is a simple matter to define the best *class-label* for a collection of objects.

Definition 1 Let S be a set of nouns, let $\mathcal{H} = \bigcup_{w \in S} H(w)$ be the set of hypernyms of S and let $\alpha(w, h)$ be an affinity score function as defined in equation (1). The *best class-label* $h_{\max}(S)$ for S is the node $h_{\max} \in \mathcal{H}$ with the highest total affinity score summed over all the members of S, so $h_{\max}$ is the node which gives the maximum score

$$\max_{h \in \mathcal{H}} \sum_{w \in S} \alpha(w, h).$$

Since $\mathcal{H}$ is determined by S, $h_{\max}$ is solely determined by the set S and the affinity score α.

In the event that $h_{\max}$ is not unique, it is customary to take the most specific class-label available.

Example

A particularly simple example of this kind of algorithm is used by Hearst and Schütze (1993). First they partition the WordNet taxonomy into a number of disjoint sets which are used as class-labels. Thus each concept has a single 'hypernym', and the 'affinity-score' between a word w and a class h is simply the set membership function, $\alpha(w, h) = 1$ if $w \in h$ and 0 otherwise. A collection of words is assigned a class-label by majority voting.

3.1 Ambiguity

In theory, rather than a class-label for related strings, we would like one for related meanings — the concepts to which the strings refer. To implement this for a set of words, we alter our affinity score function α as follows. Let $C(w)$ be the set of concepts to which the word w

could refer. (So each $c \in C$ is a possible sense of w.) Then

$$\alpha(w,h) = \max_{c \in C(w)} \begin{cases} f(\text{dist}(c,h)) & \text{if } h \in H(c) \\ -g(w,c) & \text{if } h \notin H(c), \end{cases} \quad (2)$$

This implies that the 'preferred-sense' of w with respect to the possible subsumer h is the sense closest to h. In practice, our class-labelling algorithm implements this preference by computing the affinity score $\alpha(c,h)$ for all $c \in C(w)$ and only using the best match. This selective approach is much less noisy than simply averaging the probability mass of the word over each possible sense (the technique used in (Li and Abe, 1998), for example).

3.2 Choice of scoring functions for the class-labelling algorithm

The precise choice of class-labelling algorithm depends on the functions f and g in the affinity score function α of equation (2). There is some tension here between being correct and being informative: 'correct' but uninformative class-labels (such as *entity, something*) can be obtained easily by preferring nodes high up in the hierarchy, but since our goal in this work was to classify unknown words in an informative *and* accurate fashion, the functions f and g had to be chosen to give an appropriate balance. After a variety of heuristic tests, the function f was chosen to be

$$f = \frac{1}{\text{dist}(w,h)^2},$$

where for the distance function $\text{dist}(w,h)$ we chose the computationally simple method of counting the number of taxonomic levels between w and h (inclusively to avoid dividing by zero). For the penalty function g we chose the constant $g = 0.25$.

The net effect of choosing the reciprocal-distance-squared and a small constant penalty function was that hypernyms close to the concept in question received magnified credit, but possible class-labels were not penalized too harshly for missing out a node. This made the algorithm simple and robust to noise but with a strong preference for detailed information-bearing class-labels. This configuration of the class-labelling algorithm was used in all the experiments described below.

4 Experiments and Evaluation

To test the success of our approach to placing unknown words into the WordNet taxonomy on a large and significant sample, we designed the following experiment. If the algorithm is successful at placing unknown words in the correct *new* place in a taxonomy, we would expect it to place already known words in their *current* position. The experiment to test this worked as follows.

- For a word w, find the neighbors $N(w)$ of w in WordSpace. Remove w itself from this set.
- Find the best class-label $h_{\max}(N(w))$ for this set (using Definition 1).
- Test to see if, according to WordNet, $h_{\max}$ is a hypernym of the original word w, and if so check how closely $h_{\max}$ subsumes w in the taxonomy.

Since our class-labelling algorithm gives a ranked list of possible hypernyms, credit was given for correct classifications in the top 4 places. This algorithm was tested on singular common nouns (PoS-tag *nn1*), proper nouns (PoS-tag *np0*) and finite present-tense verbs (PoS-tag *vvb*). For each of these classes, a random sample of words was selected with corpus frequencies ranging from 1000 to 250. For the noun categories, 600 words were sampled, and for the finite verbs, 420. For each word w, we found semantic neighbors with and without using part-of-speech information. The same experiments were carried out using 3, 6 and 12 neighbors: we will focus on the results for 3 and 12 neighbors since those for 6 neighbors turned out to be reliably 'somewhere in between' these two.

Results for Common Nouns

The best results for reproducing WordNet classifications were obtained for common nouns, and are summarized in Table 2, which shows the percentage of test words w which were given a class-label h which was a correct hypernym according to WordNet (so for which $h \in H(w)$). For these words for which a correct classification was found, the 'Height' columns refer to the number of levels in the hierarchy between the target word w and the class-label h. If the algorithm failed to find a class-label h which is a hypernym of w, the result was counted as 'Wrong'. The 'Missing' column records the number of words in the sample which are not in WordNet at all.

The following trends are apparent. For finding any correct class-label, the best results were obtained by taking 12 neighbors and using part-of-speech information, which found a correct classification for $485/591 = 82\%$ of the common nouns that were included in WordNet. This compares favorably with previous experiments, though as stated earlier it is difficult to be sure we are comparing like with like. Finding the hypernym which immediately subsumes w (with no intervening nodes) exactly reproduces a classification given by WordNet, and as such was taken to be a complete success. Taking fewer neighbors and using PoS-information both improved this success rate, the best accuracy obtained being $86/591 = 15\%$. However, this configuration actually gave the *worst* results at obtaining a correct classification overall.

Height	1	2	3	4	5	6	7	8	9	10	Wrong	Missing
Common Nouns (sample size 600)												
3 neighbors												
With PoS	14.3	26.1	33.1	37.8	39.8	40.6	41.5	42.0	42.0	42.0	56.5	1.5
Strings only	11.8	23.3	31.3	36.6	39.6	41.1	42.1	42.3	42.3	42.3	56.1	1.5
12 neighbors												
With PoS	10.0	21.8	36.5	48.5	59.3	70.0	76.6	78.8	79.8	80.8	17.6	1.5
without PoS	8.5	21.5	33.6	46.8	57.1	66.5	72.8	74.6	75.3	75.8	22.6	1.5
Proper Nouns (sample size 600)												
3 neighbors												
With PoS	10.6	13.8	15.5	16.5	108	18.6	18.8	18.8	19.1	19.3	25.0	55.6
Strings only	9.8	14.3	16.1	18.6	19.5	20.1	20.8	21.1	21.5	21.6	22.1	55.6
12 neighbors												
With PoS	10.5	14.5	16.3	18.1	22.0	23.8	25.5	28.0	28.5	29.3	15.0	55.6
Strings only	9.5	13.8	17.5	20.8	22.3	24.6	26.6	30.7	32.5	34.3	10.0	55.6
Verbs (sample size 420)												
3 neighbors												
With PoS	17.6	30.2	36.1	40.4	42.6	43.0	44.0	44.0	44.0	44.0	52.6	3.3
Strings only	24.7	39.7	43.3	45.4	47.1	48.0	48.3	48.8	49.0	49.0	47.6	3.3
12 neighbors												
With PoS	19.0	36.4	43.5	48.8	52.8	54.2	55.2	55.4	55.7	55.9	40.7	3.3
Strings only	28.0	48.3	55.9	60.2	63.3	64.2	64.5	65.0	65.0	65.0	31.7	3.3

Table 2: Percentage of words which were automatically assigned class-labels which subsume them in the WordNet taxonomy, showing the number of taxonomic levels between the target word and the class-label

Height	1	2	3	4	5	6	Wrong
Common Nouns	0.799	0.905	0.785	0.858	0.671	0.671	0.569
Proper Nouns	1.625	0.688	0.350	0.581	0.683	0.430	0.529
Verbs	1.062	1.248	1.095	1.103	1.143	0.750	0.669

Table 3: Average affinity score of class-labels for successful and unsuccessful classifications

In conclusion, taking more neighbors makes the chances of obtaining some correct classification for a word w greater, but taking fewer neighbors increases the chances of 'hitting the nail on the head'. The use of part-of-speech information reliably increases the chances of correctly obtaining both exact and broadly correct classifications, though careful tuning is still necessary to obtain optimal results for either.

Results for Proper Nouns and Verbs

The results for proper nouns and verbs (also in Table 2) demonstrate some interesting problems. On the whole, the mapping is less reliable than for common nouns, at least when it comes to reconstructing WordNet as it currently stands.

Proper nouns are rightly recognized as one of the categories where automatic methods for lexical acquisition are most important (Hearst and Schütze, 1993, §4). It is impossible for a single knowledge base to keep up-to-date with all possible meanings of proper names, and this would be undesirable without considerable filtering abilities because proper names are often domain-specific.

Ih our experiments, the best results for proper nouns were those obtained using 12 neighbors, where a correct classification was found for $206/266 = 77\%$ of the proper nouns that were included in WordNet, using no part-of-speech information. Part-of-speech information still helps for mapping proper nouns into exactly the right place, but in general degrades performance.

Several of the proper names tested are geographical, and in the BNC they often refer to regions of the British Isles which are not in WordNet. For example, *hampshire* is labelled as a *territorial division*, which as an English county it certainly is, but in WordNet *hampshire* is instead a hyponym of *domestic sheep*. For many of the proper names which our evaluation labelled as 'wrongly classified', the classification was in fact correct but a different meaning from those given in WordNet. The challenge for these situations is how to recognize when corpus methods give a correct meaning which is different from the meaning already listed in a knowledge base. Many of these meanings will be systematically related (such as the way a region is used to name an item or product from that region, as with the *hampshire* example above) by generative processes which are becoming well understood by theoretical linguists (Pustejovsky, 1995), and linguistic theory may help our statistical algorithms considerably by predicting what *sort* of new meanings we might expect a known word to assume through metonymy and systematic polysemy.

Typical first names of people such as *lisa* and *ralph* almost always have neighbors which are also first names (usually of the same gender), but these words are not represented in WordNet. This lexical category is ripe for automatic discovery: preliminary experiments using the two names above as 'seed-words' (Roark and Charniak, 1998; Widdows and Dorow, 2002) show that by taking a few known examples, finding neighbors and removing words which are already in WordNet, we can collect first names of the same gender with at least 90% accuracy.

Verbs pose special problems for knowledge bases. The usefulness of an IS_A hierarchy for pinpointing information and enabling inference is much less clear-cut than for nouns. For example, *sleeping* does entail *breathing* and *arriving* does imply *moving*, but the aspectual properties, argument structure and case roles may all be different. The more restrictive definition of *troponymy* is used in WordNet to describe those properties of verbs that are inherited through the taxonomy (Fellbaum, 1998, Ch 3). In practice, the taxonomy of verbs in WordNet tends to have fewer levels and many more branches than the noun taxonomy. This led to problems for our class-labelling algorithm — class-labels obtained for the verb *play* included *exhaust, deploy, move* and *behave*, all of which are 'correct' hypernyms according to WordNet, while possible class-labels obtained for the verb *appeal* included *keep, defend, reassert* and *examine*, all of which were marked 'wrong'. For our methods, the WordNet taxonomy as it stands appears to give much less reliable evaluation criteria for verbs than for common nouns. It is also plausible that similarity measures based upon simple co-occurence are better for modelling similarity between nominals than between verbs, an observation which is compatible with psychological experiments on word-association (Fellbaum, 1998, p. 90).

In our experiments, the best results for verbs were clearly those obtained using 12 neighbors and no part-of-speech information, for which some correct classification was found for $273/406 = 59\%$ of the verbs that were included in WordNet, and which achieved better results than those using part-of-speech information even for finding exact classifications. The shallowness of the taxonomy for verbs means that most classifications which were successful at all were quite close to the word in question, which should be taken into account when interpreting the results in Table 2.

As we have seen, part-of-speech information degraded performance overall for proper nouns and verbs. This may be because combining all uses of a particular word-form into a single vector is less prone to problems of data sparseness, especially if these word-forms are semantically related in spite of part-of-speech differences [2]. It is also plausible that discarding part-of-speech information

[2]This issue is reminiscent of the question of whether stemming improves or harms information retrieval (Baeza-Yates and Ribiero-Neto, 1999) — the received wisdom is that stemming (at best) improves recall at the expense of precision and our findings for proper nouns are consistent with this.

should improve the classification of verbs for the following reason. Classification using corpus-derived neighbors is markedly better for common nouns than for verbs, and most of the verbs in our sample (57%) also occur as common nouns in WordSpace. (In contrast, only 13% of our common nouns also occur as verbs, a reliable asymmetry for English.) Most of these noun senses are semantically related in some way to the corresponding verbs. Since using neighboring words for classification is demonstrably more reliable for nouns than for verbs, putting these parts-of-speech together in a single vector in WordSpace might be expected to *improve* performance for verbs but degrade it for nouns.

Filtering using Affinity scores

One of the benefits of the class-labelling algorithm (Definition 1) presented in this paper is that it returns not just class-labels but an affinity score measuring how well each class-label describes the class of objects in question. The affinity score turns out to be signficantly correlated with the likelihood of obtaining a successful classification. This can be seen very clearly in Table 3, which shows the average affinity score for correct class-labels of different heights above the target word, and for incorrect class-labels — as a rule, correct and informative class-labels have significantly higher affinity scores than incorrect class-labels. It follows that the affinity score can be used as an indicator of success, and so filtering out class-labels with poor scores can be used as a technique for improving accuracy.

To test this, we repeated our experiments using 3 neighbors and this time only using class-labels with an affinity score greater than 0.75, the rest being marked 'unknown'. Without filtering, there were 1143 successful and 1380 unsuccessful outcomes: with filtering, these numbers changed to 660 and 184 respectively. Filtering discarded some 87% of the incorrect labels and kept more than half of the correct ones, which amounts to at least a fourfold improvement in accuracy. The improvement was particularly dramatic for proper nouns, where filtering removed 270 out of 283 incorrect results and still retained half of the correct ones.

Conclusions

For common nouns, where WordNet is most reliable, our mapping algorithm performs comparatively well, accurately classifying several words and finding some correct information about most others. The optimum number of neighbors is smaller if we want to try for an exact classification and larger if we want information that is broadly reliable. Part-of-speech information noticeably improves the process of both broad and narrow classification. For proper names, many classifications are correct, and many which are absent or incorrect according to WordNet are in fact correct meanings which should be added to the knowledge base for (at least) the domain in question. Results for verbs are more difficult to interpret: reasons for this might include the shallowness and breadth of the WordNet verb hierarchy, the suitability of our WordSpace similarity measure, and many theoretical issues which should be taken into account for a successful approach to the classification of verbs.

Filtering using the affinity score from the class-labelling algorithm can be used to dramatically increase performance.

5 Related work and future directions

The experiments in this paper describe one combination of algorithms for lexical acquisition: both the finding of semantic neighbors and the process of class-labelling could take many alternative forms, and an exhaustive evaluation of such combinations is far beyond the scope of this paper. Various mathematical models and distance measures are available for modelling semantic proximity, and more detailed linguistic preprocessing (such as chunking, parsing and morphology) could be used in a variety of ways. As an initial step, the way the granularity of part-of-speech classification affects our results for lexical acquistion will be investigated. The class-labelling algorithm could be adapted to use more sensitive measures of distance (Budanitsky and Hirst, 2001), and correlations between taxonomic distance and WordSpace similarity used as a filter.

The coverage and accuracy of the initial taxonomy we are hoping to enrich has a great influence on success rates for our methods as they stand. Since these are precisely the aspects of the taxonomy we are hoping to improve, this raises the question of whether we can use automatically obtained hypernyms as well as the hand-built ones to help classification. This could be tested by randomly removing many nodes from WordNet before we begin, and measuring the effect of using automatically derived classifications for some of these words (possibly those with high confidence scores) to help with the subsequent classification of others.

The use of semantic neighbors and class-labelling for computing with meaning go far beyond the experimental set up for lexical acquisition described in this paper — for example, Resnik (1999) used the idea of a most informative subsuming node (which can be regarded as a kind of class-label) for disambiguation, as did Agirre and Rigau (1996) with the conceptual density algorithm. Taking a whole domain as a 'context', this approach to disambiguation can be used for lexical tuning. For example, using the Ohsumed corpus of medical abstracts, the top few neighbors of *operation* are *amputation, disease, therapy* and *resection*. Our algorithm gives *medical care, medical aid* and *therapy* as possible class-labels for this set, which successfully picks out the sense

of *operation* which is most important for the medical domain.

The level of detail which is appropriate for defining and grouping terms depends very much on the domain in question. For example, the immediate hypernyms offered by WordNet for the word *trout* include

> fish, foodstuff, salmonid, malacopterygian, teleost fish, food fish, saltwater fish

Many of these classifications are inappropriately fine-grained for many circumstances. To find a degree of abstraction which is suitable for the way *trout* is used in the BNC, we found its semantic neighbors which include *herring swordfish turbot salmon tuna*. The highest-scoring class-labels for this set are

2.911	saltwater fish
2.600	food fish
1.580	fish
1.400	scombroid, scombroid
0.972	teleost fish

The preferred labels are the ones most humans would answer if asked what a *trout* is. This process can be used to select the concepts from an ontology which are appropriate to a particular domain in a completely unsupervised fashion, using only the documents from that domain whose meanings we wish to describe.

Demonstration

Interactive demonstrations of the class-labelling algorithm and WordSpace are available on the web at `http://infomap.stanford.edu/classes` and `http://infomap.stanford.edu/webdemo`. An interface to WordSpace incorporating the part-of-speech information is currently under consideration.

Acknowledgements

This research was supported in part by the Research Collaboration between the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University, and by EC/NSF grant IST-1999-11438 for the MUCHMORE project.

References

E. Agirre and G. Rigau. 1996. Word sense disambiguation using conceptual density. In *Proceedings of COLING'96*, pages 16–22, Copenhagen, Denmark.

Enrique Alfonseca and Suresh Manandhar. 2001. Improving an ontology refinement method with hyponymy patterns. In *Third International Conference on Language Resources and Evaluation*, pages 235–239, Las Palmas, Spain.

Ricardo Baeza-Yates and Berthier Ribiero-Neto. 1999. *Modern Information Retrieval*. Addison Wesley / ACM Press.

A. Budanitsky and G. Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA. NAACL.

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

J. Firth. 1957. A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis, Philological Society, Oxford*, reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

Marti Hearst and Hinrich Schütze. 1993. Customizing a lexicon to better suit a computational task. In *ACL SIGLEX Workshop*, Columbus, Ohio.

Thomas Landauer and Susan Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition. *Psychological Review*, 104(2):211–240.

Hang Li and Naoki Abe. 1998. Generalizing case frames using a thesaurus and the mdl principle. *Computational Linguistics*, 24(2):217–244.

Dekang Lin. 1999. Automatic identification of non-compositional phrases. In *ACL:1999*, pages 317–324.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT press, Cambridge, MA.

Philip Resnik. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of artificial intelligence research*, 11:93–130.

Ellen Riloff and Jessica Shepherd. 1997. A corpus-based approach for building semantic lexicons. In Claire Cardie and Ralph Weischedel, editors, *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124. Association for Computational Linguistics, Somerset, New Jersey.

Brian Roark and Eugene Charniak. 1998. Noun-phrase co-occurence statistics for semi-automatic semantic lexicon construction. In *COLING-ACL*, pages 1110–1116.

Dominic Widdows and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *19th International Conference on Computational Linguistics*, pages 1093–1099, Taipei, Taiwan, August.

Dominic Widdows, Beate Dorow, and Chiu-Ki Chan. 2002. Using parallel corpora to enrich multilingual lexical resources. In *Third International Conference on Language Resources and Evaluation*, pages 240–245, Las Palmas, Spain, May.