Supersense Tagging of Unknown Nouns using Semantic Similarity

James R. Curran
School of Information Technologies
University of Sydney
NSW 2006, Australia
james@it.usyd.edu.au

Abstract

The limited coverage of lexical-semantic resources is a significant problem for NLP systems which can be alleviated by automatically classifying the unknown words. *Supersense tagging* assigns unknown nouns one of 26 broad semantic categories used by lexicographers to organise their manual insertion into WORDNET. Ciaramita and Johnson (2003) present a tagger which uses synonym set glosses as annotated training examples. We describe an unsupervised approach, based on vector-space similarity, which does not require annotated examples but significantly outperforms their tagger. We also demonstrate the use of an extremely large shallow-parsed corpus for calculating vector-space semantic similarity.

1 Introduction

Lexical-semantic resources have been applied successful to a wide range of Natural Language Processing (NLP) problems ranging from collocation extraction (Pearce, 2001) and class-based smoothing (Clark and Weir, 2002), to text classification (Baker and McCallum, 1998) and question answering (Pasca and Harabagiu, 2001). In particular, WORDNET (Fellbaum, 1998) has significantly influenced research in NLP.

Unfortunately, these resource are extremely time-consuming and labour-intensive to manually develop and maintain, requiring considerable linguistic and domain expertise. Lexicographers cannot possibly keep pace with language evolution: sense distinctions are continually made and merged, words are coined or become obsolete, and technical terms migrate into the vernacular. Technical domains, such as medicine, require separate treatment since common words often take on special meanings, and a significant proportion of their vocabulary does not overlap with everyday vocabulary. Burgun and Bodenreider (2001) compared an alignment of WORDNET with the UMLS medical resource and found only a very small degree of overlap. Also, lexical-semantic resources suffer from:

bias towards concepts and senses from particular topics. Some specialist topics are better covered in WORDNET than others, e.g. dog has finer-grained distinctions than cat and worm although this does not reflect finer distinctions in reality;

limited coverage of infrequent words and senses. Ciaramita and Johnson (2003) found that common nouns missing from WORDNET 1.6 occurred every 8 sentences in the BLLIP corpus. By WORDNET 2.0, coverage has improved but the problem of keeping up with language evolution remains difficult.

consistency when classifying similar words into categories. For instance, the WORDNET lexicographer file for ionosphere (location) is different to exosphere and stratosphere (object), two other layers of the earth's atmosphere.

These problems demonstrate the need for automatic or semi-automatic methods for the creation and maintenance of lexical-semantic resources. Broad semantic classification is currently used by lexicographers to organise the manual insertion of words into WORDNET, and is an experimental precursor to automatically inserting words directly into the WORDNET hierarchy. Ciaramita and Johnson (2003) call this *supersense tagging* and describe a multi-class perceptron tagger, which uses WORDNET's hierarchical structure to create many annotated training instances from the synset glosses.

This paper describes an unsupervised approach to supersense tagging that does not require annotated sentences. Instead, we use vector-space similarity to retrieve a number of synonyms for each unknown common noun. The supersenses of these synonyms are then combined to determine the supersense. This approach significantly outperforms the multi-class perceptron on the same dataset based on WORDNET 1.6 and 1.7.1.

Proceedings of the 43rd Annual Meeting of the ACL, pages 26–33,
Ann Arbor, June 2005.

LEX-FILE	DESCRIPTION
act	acts or actions
animal	animals
artifact	man-made objects
attribute	attributes of people and objects
body	body parts
cognition	cognitive processes and contents
communication	communicative processes and contents
event	natural events
feeling	feelings and emotions
food	foods and drinks
group	groupings of people or objects
location	spatial position
motive	goals
object	natural objects (not man-made)
person	people
phenomenon	natural phenomena
plant	plants
possession	possession and transfer of possession
process	natural processes
quantity	quantities and units of measure
relation	relations between people/things/ideas
shape	two and three dimensional shapes
state	stable states of affairs
substance	substances
time	time and temporal relations

Table 1: 25 noun lexicographer files in WORDNET

2 Supersenses

There are 26 broad semantic classes employed by lexicographers in the initial phase of inserting words into the WORDNET hierarchy, called *lexicographer files* (*lex-files*). For the noun hierarchy, there are 25 lex-files and a file containing the top level nodes in the hierarchy called Tops. Other syntactic classes are also organised using lex-files: 15 for verbs, 3 for adjectives and 1 for adverbs.

Lex-files form a set of coarse-grained sense distinctions within WORDNET. For example, company appears in the following lex-files in WORDNET 2.0: group, which covers company in the social, commercial and troupe fine-grained senses; and state, which covers companionship. The names and descriptions of the noun lex-files are shown in Table 1. Some lex-files map directly to the top level nodes in the hierarchy, called *unique beginners*, while others are grouped together as hyponyms of a unique beginner (Fellbaum, 1998, page 30). For example, abstraction subsumes the lex-files attribute, quantity, relation, communication and time.

Ciaramita and Johnson (2003) call the noun lex-file classes *supersenses*. There are 11 unique beginners in the WORDNET noun hierarchy which could also be used as supersenses. Ciaramita (2002) has produced a mini-WORDNET by manually reducing the WORDNET hierarchy to 106 broad categories. Ciaramita et al. (2003) describe how the lex-files can be used as root nodes in a two level hierarchy with the WORDNET synsets appearing directly underneath.

Other alternative sets of supersenses can be created by an arbitrary cut through the WORDNET hierarchy near the top, or by using topics from a thesaurus such as Roget's (Yarowsky, 1992). These topic distinctions are coarser-grained than WORDNET senses, which have been criticised for being too difficult to distinguish even for experts. Ciaramita and Johnson (2003) believe that the key sense distinctions are still maintained by supersenses. They suggest that supersense tagging is similar to named entity recognition, which also has a very small set of categories with similar granularity (e.g. location and person) for labelling predominantly unseen terms.

Supersense tagging can provide automated or semi-automated assistance to lexicographers adding words to the WORDNET hierarchy. Once this task is solved successfully, it may be possible to insert words directly into the fine-grained distinctions of the hierarchy itself. Clearly, this is the ultimate goal, to be able to insert new terms into lexical resources, extending the structure where necessary. Supersense tagging is also interesting for many applications that use shallow semantics, e.g. information extraction and question answering.

3 Previous Work

A considerable amount of research addresses structurally and statistically manipulating the hierarchy of WORDNET and the construction of new wordnets using the concept structure from English. For *lexical FreeNet*, Beeferman (1998) adds over 350 000 collocation pairs (*trigger pairs*) extracted from a 160 million word corpus of broadcast news using mutual information. The co-occurrence window was 500 words which was designed to approximate average document length.

Caraballo and Charniak (1999) have explored determining noun specificity from raw text. They find that simple frequency counts are the most effective way of determining the parent-child ordering, achieving 83% accuracy over types of vehicle, food and occupation. The other measure they found to be successful was the entropy of the conditional distribution of surrounding words given the noun. Specificity ordering is a necessary step for building a noun hierarchy. However, this approach clearly cannot build a hierarchy alone. For instance, entity is less frequent than many concepts it subsumes. This suggests it will only be possible to add words to an existing abstract structure rather than create categories right up to the unique beginners.

Hearst and Schütze (1993) flatten WORDNET into 726 categories using an algorithm which attempts to minimise the variance in category size. These categories are used to label paragraphs with topics, effectively repeating Yarowsky's (1992) experiments using the their categories rather than Roget's thesaurus. Schütze's (1992)

WordSpace system was used to add topical links, such as between ball, racquet and game (the *tennis problem*). Further, they also use the same vector-space techniques to label previously unseen words using the most common class assigned to the top 20 synonyms for that word.

Widdows (2003) uses a similar technique to insert words into the WORDNET hierarchy. He first extracts synonyms for the unknown word using vector-space similarity measures based on Latent Semantic Analysis and then searches for a location in the hierarchy nearest to these synonyms. This same technique as is used in our approach to supersense tagging.

Ciaramita and Johnson (2003) implement a supersense tagger based on the multi-class perceptron classifier (Crammer and Singer, 2001), which uses the standard collocation, spelling and syntactic features common in WSD and named entity recognition systems. Their insight was to use the WORDNET glosses as annotated training data and massively increase the number of training instances using the noun hierarchy. They developed an efficient algorithm for estimating the model over hierarchical training data.

4 Evaluation

Ciaramita and Johnson (2003) propose a very natural evaluation for supersense tagging: inserting the extra common nouns that have been added to a new version of WORDNET. They use the common nouns that have been added to WORDNET 1.7.1 since WORDNET 1.6 and compare this evaluation with a standard cross-validation approach that uses a small percentage of the words from their WORDNET 1.6 training set for evaluation. Their results suggest that the WORDNET 1.7.1 test set is significantly harder because of the large number of abstract category nouns, e.g. communication and cognition, that appear in the 1.7.1 data, which are difficult to classify.

Our evaluation will use exactly the same test sets as Ciaramita and Johnson (2003). The WORDNET 1.7.1 test set consists of 744 previously unseen nouns, the majority of which (over 90%) have only one sense. The WORDNET 1.6 test set consists of several cross-validation sets of 755 nouns randomly selected from the BLLIP training set used by Ciaramita and Johnson (2003). They have kindly supplied us with the WORDNET 1.7.1 test set and one cross-validation run of the WORDNET 1.6 test set. Our development experiments are performed on the WORDNET 1.6 test set with one final run on the WORDNET 1.7.1 test set. Some examples from the test sets are given in Table 2 with their supersenses.

5 Corpus

We have developed a 2 billion word corpus, shallow-parsed with a statistical NLP pipeline, which is by far the largest NLP processed corpus described in published research. The corpus consists of the *British National Corpus* (BNC), the *Reuters Corpus Volume 1* (RCV1), and most of the Linguistic Data Consortium's news text collected since 1987: *Continuous Speech Recognition III* (CSR-III); *North American News Text Corpus* (NANTC); the NANTC *Supplement* (NANTS); and the ACQUAINT *Corpus*. The components and their sizes including punctuation are given in Table 3. The LDC has recently released the *English Gigaword* corpus which includes most of the corpora listed above.

WORDNET 1.6		WORDNET 1.7.1	
NOUN	SUPERSENSE	NOUN	SUPERSENSE
stock index	communication	week	time
fast food	food	buyout	act
bottler	group	insurer	group
subcompact	artifact	partner	person
advancer	person	health	state
cash flow	possession	income	possession
downside	cognition	contender	person
discounter	artifact	cartel	group
trade-off	act	lender	person
billionaire	person	planner	artifact

Table 2: Example nouns and their supersenses

CORPUS	DOCS.	SENTS.	WORDS
BNC	4 124	6.2M	114M
RCV1	806 791	8.1M	207M
CSR-III	491 349	9.3M	226M
NANTC	930 367	23.2M	559M
NANTS	942 167	25.2M	507M
ACQUAINT	1 033 461	21.3M	491M

Table 3: 2 billion word corpus statistics

We have tokenized the text using the Grok-OpenNLP tokenizer (Morton, 2002) and split the sentences using MXTerminator (Reynar and Ratnaparkhi, 1997). Any sentences less than 3 words or more than 100 words long were rejected, along with sentences containing more than 5 numbers or more than 4 brackets, to reduce noise. The rest of the pipeline is described in the next section.

6 Semantic Similarity

Vector-space models of similarity are based on the *distributional hypothesis* that similar words appear in similar contexts. This hypothesis suggests that semantic similarity can be measured by comparing the contexts each word appears in. In vector-space models each *headword* is represented by a vector of frequency counts recording the contexts that it appears in. The key parameters are the context extraction method and the similarity measure used to compare context vectors. Our approach to

vector-space similarity is based on the SEXTANT system described in Grefenstette (1994).

Curran and Moens (2002b) compared several context extraction methods and found that the shallow pipeline and grammatical relation extraction used in SEXTANT was both extremely fast and produced high-quality results. SEXTANT extracts relation tuples $(\mathrm{w}, r, \mathrm{w}')$ for each noun, where w is the headword, r is the relation type and w' is the other word. The efficiency of the SEXTANT approach makes the extraction of contextual information from over 2 billion words of raw text feasible. We describe the shallow pipeline in detail below.

Curran and Moens (2002a) compared several different similarity measures and found that Grefenstette's weighted JACCARD measure performed the best:

$$\frac{\sum \min(\mathrm{wgt}(w_1, *_r, *_{w'}), \mathrm{wgt}(w_2, *_r, *_{w'}))}{\sum \max(\mathrm{wgt}(w_1, *_r, *_{w'}), \mathrm{wgt}(w_2, *_r, *_{w'}))} \quad (1)$$

where $\mathrm{wgt}(w, r, w')$ is the weight function for relation (w, r, w'). Curran and Moens (2002a) introduced the TTEST weight function, which is used in collocation extraction. Here, the t-test compares the joint and product probability distributions of the headword and context:

$$\frac{p(w, r, w') - p(*, r, w')p(w, *, *)}{\sqrt{p(*, r, w')p(w, *, *)}} \quad (2)$$

where $*$ indicates a global sum over that element of the relation tuple. JACCARD and TTEST produced better quality synonyms than existing measures in the literature, so we use Curran and Moen's configuration for our supersense tagging experiments.

6.1 Part of Speech Tagging and Chunking

Our implementation of SEXTANT uses a maximum entropy POS tagger designed to be very efficient, tagging at around 100 000 words per second (Curran and Clark, 2003), trained on the entire Penn Treebank (Marcus et al., 1994). The only similar performing tool is the *Trigrams 'n' Tags* tagger (Brants, 2000) which uses a much simpler statistical model. Our implementation uses a maximum entropy chunker which has similar feature types to Koeling (2000) and is also trained on chunks extracted from the entire Penn Treebank using the CoNLL 2000 script. Since the Penn Treebank separates PPs and conjunctions from NPs, they are concatenated to match Grefenstette's table-based results, i.e. the SEXTANT always prefers noun attachment.

6.2 Morphological Analysis

Our implementation uses `morpha`, the Sussex morphological analyser (Minnen et al., 2001), which is implemented using `lex` grammars for both affix splitting and generation. `morpha` has wide coverage – nearly 100% against the CELEX lexical database (Minnen et al., 2001) – and is very efficient, analysing over 80 000 words per second. `morpha` often maintains sense distinctions between singular and plural nouns; for instance: spectacles is not reduced to spectacle, but fails to do so in other cases: glasses is converted to glass. This inconsistency is problematic when using morphological analysis to smooth vector-space models. However, morphological smoothing still produces better results in practice.

RELATION	DESCRIPTION
adj	noun–adjectival modifier relation
dobj	verb–direct object relation
iobj	verb–indirect object relation
nn	noun–noun modifier relation
nnprep	noun–prepositional head relation
subj	verb–subject relation

Table 4: Grammatical relations from SEXTANT

6.3 Grammatical Relation Extraction

After the raw text has been POS tagged and chunked, the grammatical relation extraction algorithm is run over the chunks. This consists of five passes over each sentence that first identify noun and verb phrase heads and then collect grammatical relations between each common noun and its modifiers and verbs. A global list of grammatical relations generated by each pass is maintained across the passes. The global list is used to determine if a word is already attached. Once all five passes have been completed this association list contains all of the noun-modifier/verb pairs which have been extracted from the sentence. The types of grammatical relation extracted by SEXTANT are shown in Table 4. For relations between nouns (nn and nnprep), we also create inverse relations (w', r', w) representing the fact that w' can modify w. The 5 passes are described below.

Pass 1: Noun Pre-modifiers

This pass scans NPs, left to right, creating adjectival (adj) and nominal (nn) pre-modifier grammatical relations (GRs) with every noun to the pre-modifier's right, up to a preposition or the phrase end. This corresponds to assuming right-branching noun compounds. Within each NP only the NP and PP heads remain unattached.

Pass 2: Noun Post-modifiers

This pass scans NPs, right to left, creating post-modifier GRs between the unattached heads of NPs and PPs. If a preposition is encountered between the noun heads, a prepositional noun (nnprep) GR is created, otherwise an appositional noun (nn) GR is created. This corresponds to assuming right-branching PP attachment. After this phrase only the NP head remains unattached.

Tense Determination

The rightmost verb in each VP is considered the head. A

VP is initially categorised as active. If the head verb is a form of *be* then the VP becomes attributive. Otherwise, the algorithm scans the VP from right to left: if an auxiliary verb form of be is encountered the VP becomes passive; if a progressive verb (except being) is encountered the VP becomes active.

Only the noun heads on either side of VPs remain unattached. The remaining three passes attach these to the verb heads as either subjects or objects depending on the voice of the VP.

Pass 3: Verb Pre-Attachment

This pass scans sentences, right to left, associating the first NP head to the left of the VP with its head. If the VP is active, a subject (subj) relation is created; otherwise, a direct object (dobj) relation is created. For example, antigen is the subject of represent.

Pass 4: Verb Post-Attachment

This pass scans sentences, left to right, associating the first NP or PP head to the right of the VP with its head. If the VP was classed as active and the phrase is an NP then a direct object (dobj) relation is created. If the VP was classed as passive and the phrase is an NP then a subject (subj) relation is created. If the following phrase is a PP then an indirect object (iobj) relation is created. The interaction between the head verb and the preposition determine whether the noun is an indirect object of a ditransitive verb or alternatively the head of a PP that is modifying the verb. However, SEXTANT always attaches the PP to the previous phrase.

Pass 5: Verb Progressive Participles

The final step of the process is to attach progressive verbs to subjects and objects (without concern for whether they are already attached). Progressive verbs can function as nouns, verbs and adjectives and once again a naïve approximation to the correct attachment is made. Any progressive verb which appears after a determiner or quantifier is considered a noun. Otherwise, it is a verb and passes 3 and 4 are repeated to attach subjects and objects.

Finally, SEXTANT collapses the nn, nnprep and adj relations together into a single broad noun-modifier grammatical relation. Grefenstette (1994) claims this extractor has a grammatical relation accuracy of 75% after manually checking 60 sentences.

7 Approach

Our approach uses voting across the known supersenses of automatically extracted synonyms, to select a supersense for the unknown nouns. This technique is similar to Hearst and Schütze (1993) and Widdows (2003). However, sometimes the unknown noun does not appear in our 2 billion word corpus, or at least does not appear frequently enough to provide sufficient contextual information to extract reliable synonyms. In these cases, our fall-back method is a simple hand-coded classifier which examines the unknown noun and makes a guess based on simple morphological analysis of the suffix. These rules were created by inspecting the suffixes of rare nouns in WORDNET 1.6. The supersense guessing rules are given in Table 5. If none of the rules match, then the default supersense artifact is assigned.

SUFFIX	EXAMPLE	SUPERSENSE
-ness	remoteness	attribute
-tion, -ment	annulment	act
-ist, -man	statesman	person
-ing, -ion	bowling	act
-ity	viscosity	attribute
-ics, -ism	electronics	cognition
-ene, -ane, -ine	arsine	substance
-er, -or, -ic, -ee, -an	mariner	person
-gy	entomology	cognition

Table 5: Hand-coded rules for supersense guessing

The problem now becomes how to convert the ranked list of extracted synonyms for each unknown noun into a single supersense selection. Each extracted synonym votes for its one or more supersenses that appear in WORDNET 1.6. There are many parameters to consider:

- how many extracted synonyms to use;
- how to weight each synonym's vote;
- whether unreliable synonyms should be filtered out;
- how to deal with polysemous synonyms.

The experiments described below consider a range of options for these parameters. In fact, these experiments are so quick to run we have been able to exhaustively test many combinations of these parameters. We have experimented with up to 200 voting extracted synonyms.

There are several ways to weight each synonym's contribution. The simplest approach would be to give each synonym the same weight. Another approach is to use the scores returned by the similarity system. Alternatively, the weights can use the ranking of the extracted synonyms. Again these options have been considered below. A related question is whether to use all of the extracted synonyms, or perhaps filter out synonyms for which a small amount of contextual information has been extracted, and so might be unreliable.

The final issue is how to deal with polysemy. Does every supersense of each extracted synonym get the whole weight of that synonym or is it distributed evenly between the supersenses like Resnik (1995)? Another alternative is to only consider unambiguous synonyms with a single supersense in WORDNET.

A disadvantage of this similarity approach is that it requires full synonym extraction, which compares the unknown word against a large number of words when, in

SYSTEM	WN 1.6	WN 1.7.1
Ciaramita and Johnson baseline	21%	28%
Ciaramita and Johnson perceptron	53%	53%
Similarity based results	**68**%	**63**%

Table 6: Summary of supersense tagging accuracies

fact, we want to calculate the similarity to a small number of supersenses. This inefficiency could be reduced significantly if we consider only very high frequency words, but even this is still expensive.

8 Results

We have used the WORDNET 1.6 test set to experiment with different parameter settings and have kept the WORDNET 1.7.1 test set as a final comparison of best results with Ciaramita and Johnson (2003). The experiments were performed by considering all possible configurations of the parameters described above.

The following voting options were considered for each supersense of each extracted synonym: the initial voting weight for a supersense could either be a constant (IDENTITY) or the similarity score (SCORE) of the synonym. The initial weight could then be divided by the number of supersenses to share out the weight (SHARED). The weight could also be divided by the rank (RANK) to penalise supersenses further down the list. The best performance on the 1.6 test set was achieved with the SCORE voting, without sharing or ranking penalties.

The extracted synonyms are filtered before contributing to the vote with their supersense(s). This filtering involves checking that the synonym's frequency and number of contexts are large enough to ensure it is reliable. We have experimented with a wide range of cutoffs and the best performance on the 1.6 test set was achieved using a minimum cutoff of 5 for the synonym's frequency and the number of contexts it appears in.

The next question is how many synonyms are considered. We considered using just the nearest unambiguous synonym, and the top 5, 10, 20, 50, 100 and 200 synonyms. All of the top performing configurations used 50 synonyms. We have also experimented with filtering out highly polysemous nouns by eliminating words with two, three or more synonyms. However, such a filter turned out to make little difference.

Finally, we need to decide when to use the similarity measure and when to fall-back to the guessing rules. This is determined by looking at the frequency and number of attributes for the unknown word. Not surprisingly, the similarity system works better than the guessing rules if it has any information at all.

The results are summarised in Table 6. The accuracy of the best-performing configurations was 68% on the WORDNET 1.6 test set with several other parameter combinations described above performing nearly as well. On the previously unused WORDNET 1.7.1 test set, our accuracy is 63% using the best system on the WORDNET 1.6 test set. By optimising the parameters on the 1.7.1 test set we can increase that to 64%, indicating that we have not excessively over-tuned on the 1.6 test set. Our results significantly outperform Ciaramita and Johnson (2003) on both test sets even though our system is unsupervised. The large difference between our 1.6 and 1.7.1 test set accuracy demonstrates that the 1.7.1 set is much harder.

	WORDNET 1.6				WORDNET 1.7.1			
SUPERSENSE	N	P	R	F	N	P	R	F
Tops	2	0	0	0	1	50	100	67
act	84	60	74	66	86	53	73	61
animal	16	69	56	62	5	33	60	43
artifact	134	61	86	72	129	57	76	65
attribute	32	52	81	63	16	44	69	54
body	8	88	88	88	5	50	40	44
cognition	31	56	45	50	41	70	34	46
communication	66	80	56	66	57	58	44	50
event	14	83	36	50	10	80	40	53
feeling	8	70	88	78	1	0	0	0
food	29	91	69	78	12	67	67	67
group	27	75	22	34	26	50	4	7
location	43	81	30	44	13	40	15	22
motive	0	0	0	0	1	0	0	0
object	17	73	47	57	13	75	23	35
person	155	76	89	82	207	81	86	84
phenomenon	3	100	100	100	9	0	0	0
plant	11	80	73	76	0	0	0	0
possession	9	100	22	36	16	78	44	56
process	2	0	0	0	9	50	11	18
quantity	12	80	33	47	5	0	0	0
relation	2	100	50	67	0	0	0	0
shape	1	0	0	0	0	0	0	0
state	21	48	48	48	28	50	39	44
substance	24	58	58	58	44	63	73	67
time	5	100	60	75	10	36	40	38
Overall	756	68	68	68	744	63	63	63

Table 7: Breakdown of results by supersense

Table 7 shows the breakdown in performance for each supersense. The columns show the number of instances of each supersense with the precision, recall and f-score measures as percentages. The most frequent supersenses in both test sets were person, attribute and act. Of the frequent categories, person is the easiest supersense to get correct in both the 1.6 and 1.7.1 test sets, followed by food, artifact and substance. This is not surprising since these concrete words tend to have very fewer other senses, well constrained contexts and a relatively high frequency. These factors are conducive for extracting reliable synonyms.

These results also support Ciaramita and Johnson's view that abstract concepts like communication, cognition and state are much harder. We would expect the location

supersense to perform well since it is quite concrete, but unfortunately our synonym extraction system does not incorporate proper nouns, so many of these words were classified using the hand-built classifier. Also, in the data from Ciaramita and Johnson all of the words are in lower case, so no sensible guessing rules could help.

9 Other Alternatives and Future Work

An alternative approach worth exploring is to create context vectors for the supersense categories themselves and compare these against the words. This has the advantage of producing a much smaller number of vectors to compare against. In the current system, we must compare a word against the entire vocabulary (over 500 000 headwords), which is much less efficient than a comparison against only 26 supersense context vectors.

The question now becomes how to construct vectors of supersenses. The most obvious solution is to sum the context vectors across the words which have each supersense. However, our early experiments suggest that this produces extremely large vectors which do not match well against the much smaller vectors of each unseen word. Also, the same questions arise in the construction of these vectors. How are words with multiple supersenses handled? Our preliminary experiments suggest that only combining the vectors for unambiguous words produces the best results.

One solution would be to take the intersection between vectors across words for each supersense (i.e. to find the common contexts that these words appear in). However, given the sparseness of the data this may not leave very large context vectors. A final solution would be to consider a large set of the *canonical attributes* (Curran and Moens, 2002a) to represent each supersense. Canonical attributes summarise the key contexts for each headword and are used to improve the efficiency of the similarity comparisons.

There are a number of problems our system does not currently handle. Firstly, we do not include proper names in our similarity system which means that location entities can be very difficult to identify correctly (as the results demonstrate). Further, our similarity system does not currently incorporate multi-word terms. We overcome this by using the synonyms of the last word in the multi-word term. However, there are 174 multi-word terms (23%) in the WORDNET 1.7.1 test set which we could probably tag more accurately with synonyms for the whole multi-word term. Finally, we plan to implement a supervised machine learner to replace the fallback method, which currently has an accuracy of 37% on the WORDNET 1.7.1 test set.

We intend to extend our experiments beyond the Ciaramita and Johnson (2003) set to include previous and more recent versions of WORDNET to compare their difficulty, and also perform experiments over a range of corpus sizes to determine the impact of corpus size on the quality of results.

We would like to move onto the more difficult task of insertion into the hierarchy itself and compare against the initial work by Widdows (2003) using latent semantic analysis. Here the issue of how to combine vectors is even more interesting since there is the additional structure of the WORDNET inheritance hierarchy and the small synonym sets that can be used for more fine-grained combination of vectors.

10 Conclusion

Our application of semantic similarity to supersense tagging follows earlier work by Hearst and Schütze (1993) and Widdows (2003). To classify a previously unseen common noun our approach extracts synonyms which vote using their supersenses in WORDNET 1.6. We have experimented with several parameters finding that the best configuration uses 50 extracted synonyms, filtered by frequency and number of contexts to increase their reliability. Each synonym votes for each of its supersenses from WORDNET 1.6 using the similarity score from our synonym extractor.

Using this approach we have significantly outperformed the supervised multi-class perceptron Ciaramita and Johnson (2003). This paper also demonstrates the use of a very efficient shallow NLP pipeline to process a massive corpus. Such a corpus is needed to acquire reliable contextual information for the often very rare nouns we are attempting to supersense tag. This application of semantic similarity demonstrates that an unsupervised methods can outperform supervised methods for some NLP tasks if enough data is available.

Acknowledgements

We would like to thank Massi Ciaramita for supplying his original data for these experiments and answering our queries, and to Stephen Clark and the anonymous reviewers for their helpful feedback and corrections. This work has been supported by a Commonwealth scholarship, Sydney University Travelling Scholarship and Australian Research Council Discovery Project DP0453131.

References

L. Douglas Baker and Andrew McCallum. 1998. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, Australia.

Doug Beeferman. 1998. Lexical discovery with an enriched semantic network. In *Proceedings of the Workshop on Usage*

of WordNet in Natural Language Processing Systems, pages 358–364, Montréal, Québec, Canada.

Thorsten Brants. 2000. TnT - a statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 224–231, Seattle, WA USA.

Anita Burgun and Olivier Bodenreider. 2001. Comparing terms, concepts and semantic classes in WordNet and the Unified Medical Language System. In *Proceedings of the Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, pages 77–82, Pittsburgh, PA USA.

Sharon A. Caraballo and Eugene Charniak. 1999. Determining the specificity of nouns from text. In *Proceedings of the Joint ACL SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, College Park, MD USA.

Massimiliano Ciaramita and Mark Johnson. 2003. Supersense tagging of unknown nouns in WordNet. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 168–175, Sapporo, Japan.

Massimiliano Ciaramita, Thomas Hofmann, and Mark Johnson. 2003. Hierarchical semantic classification: Word sense disambiguation with world knowledge. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico.

Massimiliano Ciaramita. 2002. Boosting automatic lexical acquisition with morphological information. In *Proceedings of the Workshop on Unsupervised Lexical Acquisition*, pages 17–25, Philadelphia, PA, USA.

Stephen Clark and David Weir. 2002. Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206, June.

Koby Crammer and Yoram Singer. 2001. Ultraconservative online algorithms for multiclass problems. In *Proceedings of the 14th annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*, pages 99–115, Amsterdam, The Netherlands.

James R. Curran and Stephen Clark. 2003. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 91–98, Budapest, Hungary.

James R. Curran and Marc Moens. 2002a. Improvements in automatic thesaurus extraction. In *Proceedings of the Workshop on Unsupervised Lexical Acquisition*, pages 59–66, Philadelphia, PA, USA.

James R. Curran and Marc Moens. 2002b. Scaling context space. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 231–238, Philadelphia, PA, USA.

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

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Boston, MA USA.

Marti A. Hearst and Hinrich Schütze. 1993. Customizing a lexicon to better suit a computational task. In *Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text*, pages 55–69, Columbus, OH USA.

Rob Koeling. 2000. Chunking with maximum entropy models. In *Proceedings of the 4th Conference on Computational Natural Language Learning and of the 2nd Learning Language in Logic Workshop*, pages 139–141, Lisbon, Portugal.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Tom Morton. 2002. Grok tokenizer. *Grok OpenNLP toolkit*.

Marius Pasca and Sanda M. Harabagiu. 2001. The informative role of WordNet in open-domain question answering. In *Proceedings of the Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, pages 138–143, Pittsburgh, PA USA.

Darren Pearce. 2001. Synonymy in collocation extraction. In *Proceedings of the Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, pages 41–46, Pittsburgh, PA USA.

Philip Resnik. 1995. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, Montreal, Canada.

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 16–19, Washington, D.C. USA.

Hinrich Schütze. 1992. Context space. In *Intelligent Probabilistic Approaches to Natural Language*, number FS-92-04 in Fall Symposium Series, pages 113–120, Stanford University, CA USA.

Dominic Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 276–283, Edmonton, Alberta Canada.

David Yarowsky. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the 14th international conference on Computational Linguistics*, pages 454–460, Nantes, France.