Augmenting Approximate Similarity Searching with Lexical Information

James Gorman and **James R. Curran**
School of Information Technologies
University of Sydney
NSW 2006, Australia
{jgorman2,james}@it.usyd.edu.au

Abstract

Accurately representing synonymy using distributional similarity requires large volumes of data to reliably represent infrequent words. However, the naïve nearest-neighbour approach to compare context vectors extracted from large corpora scales poorly. The Spatial Approximation Sample Hierarchy (SASH) is a data-structure for performing approximate nearest-neighbour queries, and has been previously used to improve the scalability of distributional similarity searches. We add lexical semantic information from WordNet to the SASH in an attempt to improve the accuracy and efficiency of similarity searches.

1 Introduction

Lexical semantic resources and electronic thesauri are regularly used to solve NLP problems, including collocation discovery (Pearce, 2001), smoothing and estimation (Brown et al., 1992; Clark and Weir, 2001) and question answering (Pasca and Harabagiu, 2001). These use similarity relationships between words, as given in the resources, to enhance corpus-based statistics.

It is difficult to account for the needs of the many domains in which NLP techniques are now being applied and for rapid change in language use. Manual creation is expensive and time consuming, and open to the problems of bias, inconsistency and limited coverage. The assisted or automatic creation and maintenance of these resources would be of great advantage.

Much of the existing work on automatically extracting lexical semantic resources is based on the *distributional hypothesis* that *similar words appear in similar contexts*. Terms are described by collating information about their contexts in a corpus into a vector. These *context vectors* are then compared for similarity. Existing approaches differ primarily in their definition of "context", e.g. the surrounding words or the entire document, and their choice of distance metric for calculating similarity between the context vectors representing each term.

Finding synonyms using distributional similarity requires a nearest-neighbour search over the context vectors of each term. This is computationally intensive, scaling to the number of terms and the size of their context vectors. Curran and Moens (2002) have demonstrated that dramatically increasing the volume of raw input text used to extract context information significantly improves the quality of extracted synonyms. This will increase the size of the vocabulary, decreasing the efficiency of a naïve nearest-neighbour approach.

Using a data-structure such as the Spatial Approximation Sample Hierarchy (SASH; Houle and Sakuma, 2005) allows us to reduce the original $O(n)$ complexity (for an n term vocabulary) to $O(\log n)$ (Gorman and Curran, 2005).

The SASH represents the distributional space as a hierarchical directed graph in which each node is connected to several near-neighbour children, deriving its structure from the distribution of the space it represents. The SASH is searched by traversing these edges.

WordNet (Fellbaum, 1998) is an electronic lexical database. The main unit of organisation within WordNet is the synset, which is a collection of synonymous words. In the case of nouns, there is a secondary organisation based on hyponymy. The structure of WordNet was derived from a model of how humans understand language.

WordNet has been used successfully to solve NLP problems. Clark and Weir (2001) use the WordNet hierarchy to improve probability models of noun-predicate relationships. Pearce (2001) uses WordNet's synsets to improve collocation discovery. We investigate whether using WordNet can improve the accuracy or the efficiency of the SASH algorithm by informing the internal representation with gold-standard lexical semantic knowledge.

2 Measuring Distributional Similarity

We are measuring two classes of semantic relation using distributional similarity: synonymy and hy-

Proceedings of the Australasian Language Technology Workshop 2005, pages 49–56,
Sydney, Australia, December 2005.

ponymy/hypernymy (Curran, 2004). It is hard to distinguish between these two classes using distributional similarity.

Synonymy relates to the nearness of word meaning. Very few cases of true synonymy exist. Instead what exists is near-synonymy, where two words are not directly substitutable, but share some close common meaning. The distinction between loud and noisy is an example of this. They both represent the idea of high volume sound, but noisy also has a negative connotation not present in loud.

Measuring distributional similarity first requires the extraction of context information for each of the vocabulary terms from raw text. These terms are then compared for similarity using a nearest-neighbour search or clustering based on distance calculations between the statistical descriptions of their contexts.

2.1 Extraction Method

A *context relation* is defined as a tuple (w, r, w') where w is a term, which occurs in some grammatical relation r with another word w' in some sentence. We refer to the tuple (r, w') as an *attribute* of w. For example, (dog, direct-obj, walk) indicates that dog was the direct object of walk in a sentence.

Context extraction begins with a Maximum Entropy POS tagger and chunker (Ratnaparkhi, 1996). The SEXTANT relation extractor (Grefenstette, 1994) produces context relations that are then lemmatised using the Minnen et al. (2000) morphological analyser. The relations for each term are collected together and counted, producing a vector of attributes and their frequencies in the corpus.

The syntactic contexts that are extracted by SEXTANT are:

1. term is the subject of a verb
2. term is the (direct/indirect) object of a verb
3. term is modified by a noun or adjective
4. term is modified by a prepositional phrase

2.2 Measures and Weights

Both nearest-neighbour and cluster analysis methods require a distance measure to calculate the similarity between context vectors. Curran (2004) decomposes this into *measure* and *weight* functions. The *measure* function calculates the similarity between two weighted context vectors and the *weight* function calculates a weight from the raw frequency information for each context relation.

For these experiments we use the JACCARD (1) measure and the TTEST (2) weight functions, as Curran (2004) found them to have the best performance in his comparison of many distance measures.

$$\frac{\sum_{(r,w')} \min(\mathrm{wgt}(\mathrm{w}_m, r, w'), \mathrm{wgt}(\mathrm{w}_n, r, w'))}{\sum_{(r,w')} \max(\mathrm{wgt}(\mathrm{w}_m, r, w'), \mathrm{wgt}(\mathrm{w}_n, r, w'))} \tag{1}$$

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

2.3 Nearest-neighbour search

The simplest algorithm for finding synonyms is a k-nearest-neighbour (k-NN) search, which involves pair-wise vector comparison of the target term with every term in the vocabulary. Given an n term vocabulary and up to m attributes for each term, the asymptotic time complexity of nearest-neighbour search is $O(n^2m)$. This is very expensive, with even a moderate vocabulary making the use of huge datasets infeasible. It is for this reason that the SASH data-structure is used to reduce the time complexity.

3 The SASH

The SASH approximates a k-NN search by precomputing some near neighbours for each node (terms in our case). This produces multiple paths between terms, allowing the SASH to shape itself to the data set (Houle, 2003). The following description is adapted from Houle and Sakuma (2005).

The SASH is a directed, edge-weighted graph with the following properties (see Figure 1):

- Each term corresponds to a unique node.
- The nodes are arranged into a hierarchy of levels, with the bottom level containing $\frac{n}{2}$ nodes and the top containing a single root node. Each level, except the top, will contain half as many nodes as the level below. These are numbered from 1 (top) to h.
- Edges between nodes are linked from consecutive levels. Each node will have at most p *parent* nodes in the level above, and c *child* nodes in the level below.
- Every node must have at least one parent so that all nodes are reachable from the root.

Construction begins with the nodes being randomly distributed between the levels. The SASH is then constructed iteratively by each node finding its closest p parents in the level above. The parent will keep the closest c of these children, forming edges in the graph, and reject the rest. Any nodes without parents after being rejected are then assigned as children of the nearest node in the previous level with fewer than c children.

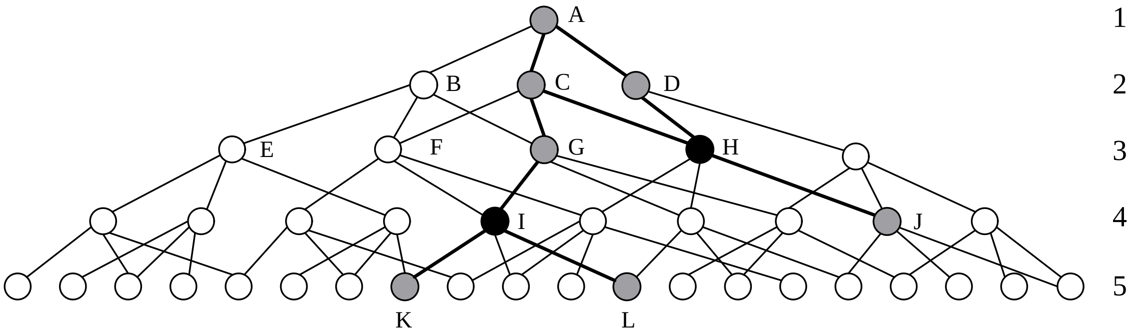

Figure 1: A SASH, where $p = 2$, $c = 3$ and $k = 2$

Searching is performed by finding the k nearest nodes at each level, which are added to a set of near nodes. To limit the search, only those nodes whose parents were found to be nearest at the previous level are searched. The k closest nodes from the set of near nodes are then returned. The search complexity is $ck \log_2 n$.

In Figure 1, the filled nodes demonstrate a search for the near-neighbours of some node q, using $k = 2$. Our search begins with the root node A. As we are using $k = 2$, we must find the two nearest children of A using our similarity measure. In this case, C and D are closer than B. We now find the closest two children of C and D. E is not checked as it is only a child of B. All other nodes are checked, including F and G, which are shared as children by B and C. From this level we chose G and H. We then consider the fourth and fifth levels similarly.

At this point we now have the list of near nodes A, C, D, G, H, I, J, K and L. From this we chose the two nodes nearest q: H and I marked in black. These are returned as the near-neighbours of q.

k can be varied at each level to force a larger number of elements to be tested at the base of the SASH using, for instance, the equation:

$$k_i = \max\{\, k^{1-\frac{h-i}{\log_2 n}}, \frac{1}{2}pc \,\} \qquad (3)$$

This changes our search complexity to:

$$\frac{k^{1+\frac{1}{\log_2 n}}}{k^{\frac{1}{\log_2 n}}-1} + \frac{pc^2}{2}\log_2 n \qquad (4)$$

(Houle and Sakuma, 2005). We use this geometric function in our experiments.

4 WordNet

Within WordNet (Fellbaum, 1998), words are divided into four syntactic categories: noun, verb, adjective and adverb. Each of these categories has a different structure, representing their use. We are only concerned with nouns in these experiments and, when referring to WordNet, we only refer to this part of it.

The key building block of WordNet is the *synset*: a set of synonymous terms. Words in a synset may not be fully interchangeable, but are in at least some contexts. Because words are organised by concept, polysemous words will appear in several synsets.

Synsets are arranged in a hierarchy based on hyponymic relations. Those near the root are more general, and those near the leaves are more specific.

WordNet 2.1 consists of 117,097 unique terms in 81,426 synsets. Of these terms 15,776 are polysemous, yielding a total of 145,104 word-sense pairs. Our experimental corpus consists of 246,067 unique terms, of which 88,925 remain after a frequency cutoff of 5 is applied. 22,537 terms occur in both WordNet and our corpus, yielding 32,057 senses.

A coarse-grained sense distinction is made by 25 lexicographer files (see Table 1). Each of these represent distinct conceptual and lexical domains and were selected to cover all possible English nouns. These map to the top most synsets in the WordNet hierarchy, either uniquely or as hyponyms.

Table 1 also show the proportion of WordNet covered by each domain (by type), and the proportion of the terms in both the BNC and WordNet in each domain (by token from the BNC). We represent our corpus statistics by token as this is indicative of how reliable the context information is for each domain. Where a term appears in several domains, its count is divided by the number of domains and spread evenly between them, following the Resnik (1995) uniform mass splitting strategy.

WordNet itself can be used to measure semantic similarity. Budanitsky and Hirst (2001) found the method proposed by Jiang and Conrath (1997) to be the most successful in malapropism detection. They used information content to measure the conditional probability of finding a child synset given a parent synset.

Leacock and Chodorow (1998) measure the log

act, activity	7.0%	11.4%	natural object	1.8%	1.7%
animal, fauna	10.9%	4.5%	natural phenomenon	0.8%	0.8%
artifact	12.3%	16.1%	person, human being	13.8%	14.7%
attribute	3.4%	6.5%	plant, flora	12.6%	2.6%
body	2.8%	2.5%	possession	1.1%	1.0%
cognition, knowledge	3.3%	4.9%	process	0.9%	1.3%
communication	6.3%	7.8%	quantity, amount	1.4%	2.0%
event, happening	1.2%	2.2%	relation	0.5%	0.6%
feeling, emotion	0.6%	1.3%	shape	0.4%	0.6%
food	2.8%	2.7%	state	4.4%	5.2%
group, grouping	3.0%	2.2%	substance	3.7%	4.7%
location	3.8%	1.1%	time	1.3%	1.1%
motivation, motive	0.1%	0.1%			

Table 1: 25 lexicographer files (Fellbaum, 1998)

of the path distance between two synsets, scaled by the overall depth of the hierarchy. This performed nearly as well as Jiang and Conrath's method.

5 Evaluation

Our evaluation uses a combination of three electronic thesauri: the Macquarie (Bernard, 1990), Roget's (Roget, 1911) and Moby (Ward, 1996) thesauri. It is possible to use precision and recall measures to evaluate the quality of the extracted thesaurus. To help overcome the problems of direct comparisons we use several measures of system performance: direct matches (DIRECT), inverse rank (INVR), and precision of the top n synonyms (P(n)), for n = 1, 5 and 10.

INVR is the sum of the inverse rank of each matching synonym, e.g. matches at ranks 3, 5 and 28 give an inverse rank score of $\frac{1}{3} + \frac{1}{5} + \frac{1}{28}$. With at most 100 synonyms, the maximum INVR score is 5.187. P(n) is the percentage of matching synonyms in the top n extracted synonyms.

The same 300 single-word nouns were used for the evaluation as used by Curran (2004) for his large scale evaluation. These were chosen randomly from WordNet such that they covered a range over the following properties:

frequency Penn Treebank and BNC frequencies

number of senses WordNet and Macquarie senses

specificity depth in the WordNet hierarchy

concreteness distribution across WordNet subtrees

For each of these terms, the closest 100 terms and their similarity score were extracted.

6 Experiments

The contexts were extracted from the non-speech portion of the British National Corpus (Burnard, 1995). All experiments used the JACCARD measure function, the TTEST weight function and a cut-off frequency of 5. The SASH was constructed using the geometric equation for k_i described in Section 3.

The values 1–4, 2–8, 4–16, 8–32 and 16–64 were chosen for number of parents (p) and children (c) in the SASH, giving are range of branching factors to test the balance between *sparseness* and *bushiness*.

As in Gorman and Curran (2005), we use the brute force k-NN search (NAIVE) as our base-line for all our experiments. We also reproduce the results for the fully random distribution (RANDOM), when ordered by frequency (SORT) and when *folded* about some number of relations (FOLD$\mathcal{M}$).

RANDOM is consistent with the original design of the SASH. In accordance with Zipf's law (Zipf, 1949), the majority of the terms have low frequencies, and comparisons with these low frequency terms are unreliable (Curran and Moens, 2002), SORT forces high frequency terms towards the root, producing more accurate results by providing more reliable initial search paths.

Unfortunately, these more reliable search paths are also more expensive to calculate. To mitigate this, FOLD$\mathcal{M}$ chooses *more* accurate initial paths, rather than *most* accurate paths. For each term, if its number of relations m_i is greater than some chosen number of relations $\mathcal{M}$, it is given a new ranking based on the score $\frac{\mathcal{M}^2}{m_i}$. Otherwise its ranking based on its number of relations. This has the effect of pushing very high and very low frequency terms away form the root. The folding points this was tested for were 500, 1000 and 1500.

7 Integrating WordNet

Integrating information from WordNet produces much more complicated sorting schemes. The most direct method of using WordNet would be to *use* the WordNet hierarchy as the top levels of the SASH. Those terms present in our vocabulary and in WordNet would be inserted into the SASH in the same order and with the same linkages as given by Word-

Dist	c	Direct	P(1)	P(5)	P(10)	InvR	Time
Naive		5.29	60%	47%	39%	1.72	12217ms
Random	8	4.93	61%	47%	39%	1.71	520ms
Random	16	5.23	60%	48%	39%	1.73	872ms
Random	32	5.30	60%	47%	39%	1.74	1899ms
Sort	8	4.89	62%	47%	39%	1.71	317ms
Sort	16	5.30	61%	48%	39%	1.75	677ms
Sort	32	5.32	60%	48%	39%	1.74	1709ms

Table 2: Evaluation of random and fully sorted distributions

Net. Those terms in our vocabulary and not in WordNet would then be inserted into levels below the already linked terms, and then normal SASH building process would link them.

This method is very different to the original design of the SASH. Even when we order by frequency or number of relations, the ordering of semantic relations is still random because synonymy is not a function of frequency. The SASH relies on this randomness to cluster the terms successfully.

Despite the paths in WordNet being between semantically similar terms, the success of this method is doubtful. Many terms at the top of the hierarchy, where searches begin, will not produce reliable measurements. Some, such as thing, are too general to narrow a search. Others, such as psychological feature, will occur with such a low frequency as to make measurement unreliable.

The most specific terms at the bottom of the WordNet hierarchy will have those terms not in WordNet as children. These specific terms are likely to have a lower frequency than terms in the middle of the hierarchy. The low frequency WordNet terms will produce less accurate paths when they find their children during construction, resulting in unreliable searches for terms not in WordNet. The fixed structure of the WordNet paths will also reduce the ability to find new similarities within WordNet as the paths to these will not exist.

Rather than using the knowledge provided by the synsets and hyponymy relations directly, we use the knowledge that both WordNet and the SASH arrange terms as a graph. From WordNet, use additional knowledge from the 25 lexicographer files covering distinct conceptual domains (Table 1).

An analysis of the terms occuring in both WordNet and our corpus shows an uneven distribution. The act, artifact and person domains each represent 10–15% of these terms, while the motivation, relation and shape domains represent less than 1%. When randomly distributed, there will be many more high frequency domains represented at the top of the SASH. The initial paths formed at the top of a SASH determine the accruacy of searches. If the initial path is innaccurate, then there is little chance of finding correct near-neighbours. If a domain is not represented at the top of the SASH, any searches for terms in that domain will first have to pass through domains with which they have little similarity. These paths are likely to be inaccurate, reducing the accuracy for the whole search.

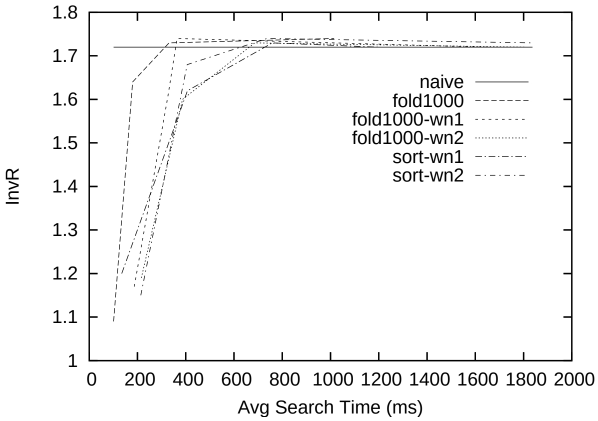

Figure 2: InvR against average search time

We want all conceptual domains represented evenly at the top levels of the SASH, without overly affecting the distribution of the terms themselves. Both Sort and Fold$\mathcal{M}$ improve the performance, preserving the distribution of terms, but do not guarantee the even distribution of the domains. We want to ensure this even distribution.

To combine information from WordNet and our existing sorting techniques, we split our vocabulary according to membership of domains. This provides us with 25 lists of terms that appear in WordNet, and single a list of those that do not.

Each of these lists are then sorted by one of the sorting schemes (Random, Sort or Fold$\mathcal{M}$). The lists are then merged by taking the current topmost term from each list and inserting it into a single list that will be used to create the SASH. For polysemous terms appearing in several lists, the list with the highest sorting is used.

Those terms not appearing in WordNet are treated in two ways. The first (WN1) is to treat them as

DIST	*c*	DIRECT	P(1)	P(5)	P(10)	INVR	Time
NAIVE		5.29	60%	47%	39%	1.72	12217ms
FOLD500	8	4.24	60%	45%	35%	1.60	185ms
FOLD500	16	5.15	62%	48%	39%	1.75	336ms
FOLD500	32	5.30	60%	48%	39%	1.74	961ms
FOLD1000	8	4.43	60%	46%	37%	1.64	180ms
FOLD1000	**16**	**5.21**	**61%**	**48%**	**39%**	**1.73**	**331ms**
FOLD1000	32	5.31	60%	48%	39%	1.74	1015ms
FOLD1500	8	4.43	59%	45%	37%	1.62	236ms
FOLD1500	16	5.21	61%	48%	39%	1.74	366ms
FOLD1500	32	5.31	60%	48%	39%	1.74	1157ms

Table 3: Evaluation of folded distributions

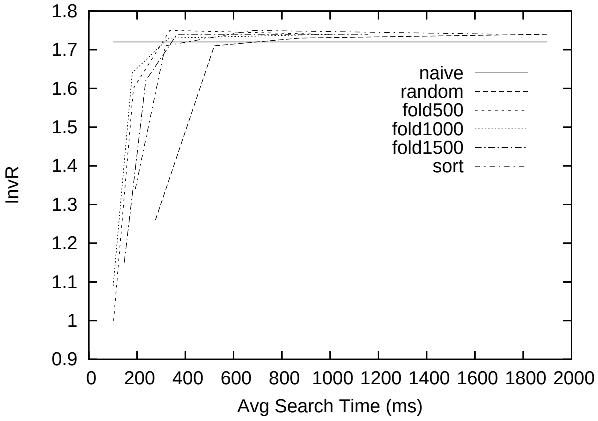

Figure 3: INVR against average search time

a twenty-sixth lexical category and merge them as such. The second (WN2) is to place these terms after those that appear in WordNet. This is more in the spirit of the method, as those terms not in WordNet are not all of a single domain.

Although it would seem that only having one quarter of the terms in the SASH arranged by domain would be too few to have an effect, this represents the top thirteen levels of our SASH, from at total of fifteen. As noise is more significant in initial path formation, the effect of changing the distribution at the top of the SASH has more affect than changing it at the bottom.

8 Results

Figure 2 plots the trade-off between accuracy and efficiency after we have introduced WordNet information into the SASH, using values of *c* between 4 and 64. The initial sharp increase in efficiency is for values of *c* from 4 to 8. We see knee points between 400 and 600ms for the WordNet distributions, and 300ms for FOLD1000, when *c* is between 8 and 16. After the INVR exceeds NAIVE, we have a long tail where INVR converges on NAIVE as the search time increases. What is most interesting in the *sharp* knee of FOLD1000-WN1. From performing worse than FOLD1000, it increases sharply to an equivalent performance, then converges to a an equivalent INVR to NAIVE.

Figure 3 plots the trade-off between accuracy and efficiency for RANDOM, SORT and FOLD1000 using INVR and search time, again using the values of *c* between 4 and 64. This can be contrasted with Figure 2. Again we have an initial sharp increase in efficiency, and a long tail converging to NAIVE. All the SASH distributions have their knee points at around 300-500ms, when *c* is between 8 and 16.

Table 2 presents the results for the original NAIVE, RANDOM and SORT experiments. These have been run using an improved implementation of the SASH from that used in Gorman and Curran (2005). Only the results for $c = 8$, 16 and 32 are shown, as these span the knee point. SORT consistently outperformed RANDOM in efficiency and outperformed RANDOM in accuracy for $c \geq 16$. Both SASH solutions outperformed NAIVE in efficiency by more than 14 times when $c = 16$. At $c = 16$, SORT produced similar results for DIRECT and outperformed in INVR by 1%. RANDOM produced a similar INVR and was outperformed in DIRECT by 1%.

Table 3 presents the results for the folded distributions. At $c = 16$, these produced accuracies equivalent to RANDOM, at twice the speed of SORT and 33 times the speed of NAIVE. FOLD1500 was the slowest, although only by 30ms, which cannot be considered significant. Its accuracy was 98% of DIRECT and equivalent INVR of NAIVE. FOLD500 has the highest INVR at 1.75, but the lowest DIRECT at 97% of NAIVE. FOLD1000 provided the best balance with the accuracy of FOLD1500 and the speed of FOLD500.

Table 4 presents the results when WordNet information is used. RANDOM-WN1 similar accuracy to, but is nearly time as fast as RANDOM. RANDOM-WN2 produces similar accuracy, but with only a minor increase in efficiency. SORT-WN1

DIST	DIRECT	P(1)	P(5)	P(10)	INVR	Time
NAIVE	5.29	60%	47%	39%	1.72	12217ms
FOLD1000	**5.21**	**61%**	**48%**	**39%**	**1.73**	**331ms**
RANDOM-WN1	5.24	59%	47%	39%	1.72	488ms
RANDOM-WN2	5.25	59%	48%	39%	1.73	773ms
SORT-WN1	5.26	59%	48%	39%	1.73	759ms
SORT-WN2	5.30	59%	48%	39%	1.74	737ms
FOLD1000-WN1	5.23	59%	48%	39%	1.73	686ms
FOLD1000-WN2	5.23	59%	48%	39%	1.73	686ms

Table 4: Evaluation of WordNet distributions

produces a similar accuracy SORT, but is slower. SORT-WN2 is also slower and suffer a minor accuracy penalty. FOLD1000-WN1 produces a similar accuracy to FOLD1000 and a similar search time. FOLD1000-WN2 produces a similar accuracy and is nearly twice as slow.

The consistent pattern in the results is that once we order by frequency or relations, any improvements in accuracy are not significant. In addition, any improvements from using WordNet information are inconsistent.

9 Analysis

The results for using the SASH without WordNet show that it provides a significant improvement over a naïve search. It is less clear whether adding the WordNet information brings further improvement.

FOLD1000-WN1 produces a result that is similar to the best results for FOLD1000. RANDOM-WN1 is much faster than RANDOM without a loss in accuracy All other results using WordNet are worse.

What we see most here is that there is no obvious pattern to the effects of adding WordNet information to the SASH. In most cases it simply degrades performance, but sometime it improves aspects of it. This occurs for both WN1 and WN2, using different base distributions. A deeper analysis is needed.

There was no general pattern where a distribution of the SASH was more accurate for some term than others except for approximately 25 terms which scored consistently lower or higher when WordNet information was used. These words were compared for polysemy, lexical file membership, depth in the hierarchy, distance, corpus frequency and number of relations. None of these provided any pattern as to identifying either high or low scoring terms.

The analysis of the SASH covered both the construction and the searches. The construction considered the distribution of terms and the number of children of each term. Term distribution was measured by calculating the proportion of terms shared between two distributions for a certain number of terms a the top the distribution. RANDOM distributions share an average of 1% of the top 1000 terms with any other distribution. Between 57% and 69% of terms were shared between distributions with and without WordNet information. Although there was a pattern following the distribution, there was none that indicated its success.

Children were counted to determine if there was a change in the bushiness of the SASH as the distribution changed. This was considered both globally and for each level of the SASH. Again trends were only indicative of the distribution. This was extended to consider the average distance to and the number of relations of each child without yielding further information.

Searches were analysed by measuring the proportion of searching done at each level. This considered the number of terms compared, the number number of relations compared and distance to the search term. This showed no trends.

Given that no trends were found indicating which broad structural and distributional changes had a positive influence, we are left to conclude that the problem lies in the way the SASH clusters particular distributions.

When used as designed the SASH is robust. Our initial distribution functions all produce stable results for various values of c and p. WordNetinformation can improve the performance of the RANDOM distribution, but our SORT and FOLD$\mathcal{M}$ ordering functions increase the stability of the data at the top of the SASH, improving results without needing additional lexical information.

10 Conclusion

We have used lexical semantic information from WordNet to inform the internal structure of the Spacial Approximation Sample Hierarchy (SASH). The SASH has shown the current methods of improving performance to be stable enough that adding this information does not provide any benefit.

That we had some positive results using WordNet, albeit inconsistently, indicates that using lexical information may still provide some improvement in accuracy or efficiency. What this information is and how it should be combined are questions that are yet to be answered. Although dismissed in its simplest form, using WordNet more directly in the SASH presents one possible direction.

We intend to further investigate using lexical semantic information to improve performance, implement other term ordering strategies, as well as further investigating the canonical vector heuristic presented in Gorman and Curran (2005).

Having set out with the aim of applying lexical knowledge to approximate distributional similarity searches, we have found that the existing methods for improving the performance of the SASH are sufficiently robust that this is unnecessary.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful feedback and corrections. This work has been supported by the Australian Research Council under Discovery Project DP0453131.

References

John R. L. Bernard, editor. 1990. *The Macquarie Encyclopedic Thesaurus*. The Macquarie Library, Sydney, Australia.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, December.

Alexander Budanitsky and Graeme Hirst. 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA, USA, 2–7 June.

Lou Burnard, editor. 1995. *Users Reference Guide British National Corpus Version 1.0*. Oxford University Computing Services, Oxford, UK.

Stephen Clark and David Weir. 2001. Class-based probability estimation using a semantic hierarchy. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 95–102, Pittsburgh, PA, USA, 2–7 June.

James Curran and Marc Moens. 2002. Scaling context space. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 231–238, Philadelphia, PA, USA, 7–12 July.

James Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh, Edinburgh, UK.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. The MIT Press, Cambridge, MA, USA.

James Gorman and James Curran. 2005. Approximate searching for distributional similarity. In *ACL-SIGLEX 2005 Workshop on Deep Lexical Acquisition*, Ann Arbor, MI, USA, 30 June.

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

Michael E. Houle and Jun Sakuma. 2005. Fast approximate similarity search in extremely high-dimensional data sets. In *Proceedings of the 21st International Conference on Data Engineering*, pages 619–630, Tokyo, Japan, 5–8 April.

Michael E. Houle. 2003. Navigating massive data sets via local clustering. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 547–552, Washington, DC, USA, 24–27 August.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of 10th International Conference on Research in Computational Linguistics*, pages 19–33, Taipei, Taiwan, 22–24 August.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In *Fellbaum, 1998*, pages 265–283.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust applied morphological generation. In *Proceedings of the 1st International Natural Language Generation Conference*, pages 201–208, Mitzpe Ramon, Israel, 12–16 June.

Marius Pasca and Sanda 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, 2–7 June.

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, 2–7 June.

Adwait Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, 17–18 May.

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, 20–25 August.

Peter Roget. 1911. *Thesaurus of English words and phrases*. Longmans, Green and Co., London, UK.

Grady Ward. 1996. *Moby Thesaurus*. Moby Project.

George K. Zipf. 1949. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, Cambridge, MA, USA.