

Scaling Context Space

James R. Curran and **Marc Moens**

Institute for Communicating and Collaborative Systems

University of Edinburgh

2 Buccleuch Place, Edinburgh EH8 9LW

United Kingdom

{jamesc, marc}@cogsci.ed.ac.uk

Abstract

Context is used in many NLP systems as an indicator of a term's syntactic and semantic function. The accuracy of the system is dependent on the quality and quantity of contextual information available to describe each term. However, the quantity variable is no longer fixed by limited corpus resources. Given fixed training time and computational resources, it makes sense for systems to invest time in extracting high quality contextual information from a fixed corpus. However, with an effectively limitless quantity of text available, extraction rate and representation size need to be considered. We use thesaurus extraction with a range of context extracting tools to demonstrate the interaction between context quantity, time and size on a corpus of 300 million words.

1 Introduction

Context plays an important role in many natural language tasks. For example, the accuracy of part of speech taggers or word sense disambiguation systems depends on the quality and quantity of contextual information these systems can extract from the training data. When predicting the sense of a word, for instance, the immediately preceding word is likely to be more important than the tenth previous word; similar observations can be made about POS taggers or chunkers. A crucial part of training these systems lies in extracting from the data high-quality contextual information, in the sense of

defining contexts that are both accurate and correlated with the information (the POS tags, the word senses, the chunks) the system is trying to extract.

The quality of contextual information is often determined by the size of the training corpus: with less data available, extracting context information for any given phenomenon becomes less reliable. However, corpus size is no longer a limiting factor: whereas up to now people have typically worked with corpora of around one million words, it has become feasible to build much larger document collections; for example, Banko and Brill (2001) report on experiments with a one billion word corpus.

When using a much larger corpus and scaling the context space, there are, however, other trade-offs to take into consideration: the size of the corpus may make it unfeasible to train some systems because of efficiency issues or hardware costs; it may also result in an unmanageable expansion of the extracted context information, reducing the performance of the systems that have to make use of this information.

This paper reports on experiments that try to establish some of the trade-offs between corpus size, processing time, hardware costs and the performance of the resulting systems. We report on experiments with a large corpus (around 300 million words). We trained a thesaurus extraction system with a range of context-extracting front-ends to demonstrate the interaction between context quality, extraction time and representation size.

2 Automatic Thesaurus Extraction

Thesauri have traditionally been used in information retrieval tasks to expand words in queries with synonymous terms (e.g. Ruge, (1997)). More re-

cently, semantic resources have also been used in collocation discovery (Pearce, 2001), smoothing and model estimation (Brown et al., 1992; Clark and Weir, 2001) and text classification (Baker and McCallum, 1998). Unfortunately, thesauri are very expensive and time-consuming to produce manually, and tend to suffer from problems of bias, inconsistency, and lack of coverage. In addition, thesaurus compilers cannot keep up with constantly evolving language use and cannot afford to build new thesauri for the many subdomains that information extraction and retrieval systems are being developed for. There is a clear need for methods to extract thesauri automatically or tools that assist in the manual creation and updating of these semantic resources.

Most existing work on thesaurus extraction and word clustering is based on the general observation that related terms will appear in similar contexts. The differences tend to lie in the way “context” is defined and in the way similarity is calculated. Most systems extract co-occurrence and syntactic information from the words surrounding the target term, which is then converted into a vector-space representation of the contexts that each target term appears in (Brown et al., 1992; Pereira et al., 1993; Ruge, 1997; Lin, 1998b). Other systems take the whole document as the context and consider term co-occurrence at the document level (Crouch, 1988; Sanderson and Croft, 1999). Once these contexts have been defined, these systems then use clustering or nearest neighbour methods to find similar terms.

Finally, some systems extract synonyms directly without extracting and comparing contextual representations for each term. Instead, these systems recognise terms within certain linguistic patterns (e.g. *X, Y and other Zs*) which associate synonyms and hyponyms (Hearst, 1992; Caraballo, 1999).

Thesaurus extraction is a good task to use to experiment with scaling context spaces. The vector-space model with nearest neighbour searching is simple, so we needn’t worry about interactions between the contexts we select and a learning algorithm (such as independence of the features). But also, thesaurus extraction is a task where success has been limited when using small corpora (Grefenstette, 1994); corpora of the order of 300 million words have already been shown to be more successful at this task (Lin, 1998b).

3 Experiments

Vector-space thesaurus extraction can be separated into two independent processes. The first step extracts the contexts from raw text and compiles them into a vector-space statistical description of the contexts each potential thesaurus term appears in.

We define a *context relation* as a tuple (w, r, w') where w is a thesaurus term, occurring in relation type r , with another word w' in the sentence. The type can be grammatical or the position of w' in a context window: the relation $(\text{dog}, \text{direct-obj}, \text{walk})$ indicates that the term `dog`, was the direct object of the verb `walk`. Often we treat the tuple (r, w') as a single unit and refer to it as an *attribute* of w . The context extraction systems used for these experiments are described in the following section.

The second step in thesaurus extraction performs clustering or nearest-neighbour analysis to determine which terms are similar based on their context vectors. Our second component is similar to Grefenstette’s SEXTANT system, which performs nearest-neighbour calculations for each pair of potential thesaurus terms. For nearest-neighbour measurements we must define a function to judge the similarity between two context vectors (e.g. the cosine measure) and a function to combine the raw instance frequencies for each context relation into weighted vector components.

SEXTANT uses a generalisation of the Jaccard measure to measure similarity. The Jaccard measure is the cardinality ratio of the intersection and union of attribute sets ($\text{atts}(w_n)$ is the attribute set for w_n):

$$\frac{|\text{atts}(w_m) \cap \text{atts}(w_n)|}{|\text{atts}(w_m) \cup \text{atts}(w_n)|} \quad (1)$$

The generalised Jaccard measure allows each relation to have a significance weight (based on word, attribute and relation frequencies) associated with it:

$$\frac{\sum_{a \in \text{atts}(w_m) \cup \text{atts}(w_n)} \min(\text{wgt}(w_m, a), \text{wgt}(w_n, a))}{\sum_{a \in \text{atts}(w_m) \cup \text{atts}(w_n)} \max(\text{wgt}(w_m, a), \text{wgt}(w_n, a))} \quad (2)$$

Grefenstette originally used the weighting function:

$$\text{wgt}(w_i, a_j) = \frac{\log_2(f(w_i, a_j) + 1)}{\log_2(n(a_j) + 1)} \quad (3)$$

where $f(w_i, a_j)$ is the frequency of the relation and $n(a_j)$ is the number of different words a_j appears in relations with.

Name	Context Description
$W(L_1R_1)$	one word to left or right
$W(L_1)$	one word to the left
$W(L_{1,2})$	one or two words to the left
$W(L_{1-3})$	one to three words to the left

Table 1: Window extractors

However, we have found that using the t-test between the joint and independent distributions of a word and its attribute:

$$\text{wgt}(w_i, a_j) = \frac{p(w_i, a_j) - p(w_i)p(a_j)}{\sqrt{p(w_i)p(a_j)}} \quad (4)$$

gives superior performance (Curran and Moens, 2002) and is therefore used for our experiments.

4 Context Extractors

We have experimented with a number of different systems for extracting the contexts for each word. These systems show a wide range in complexity of method and implementation, and hence development effort and execution time.

The simplest method we implemented extracts the occurrence counts of words within a particular window surrounding the thesaurus term. These window extractors are very easy to implement and run very quickly. The window geometries used in this experiment are listed in Table 1. Extractors marked with an asterisk, for example $W(L_1R_1^*)$, do not distinguish (within the relation type) between different positions of the word w' in the window.

At a greater level of complexity we have two shallow NLP systems which provide extra syntactic information in the extracted contexts. The first system is based on the syntactic relation extractor from SEXTANT with a different POS tagger and chunker. The SEXTANT-based extractor we developed uses a very simple Naïve Bayes POS tagger and chunker. This is very simple to implement and is extremely fast since it optimises the tag selection locally at the current word rather than performing beam or Viterbi search over the entire sentence. After the raw text has been POS tagged and chunked, the SEXTANT relation extraction algorithm is run over the text. This consists of five passes over each sentence that associate each noun with the modifiers and verbs from the syntactic contexts that it appears in.

Corpus	Sentences	Words
British National Corpus	6.2M	114M
Reuters Corpus Vol 1	8.7M	193M

Table 2: Training Corpora Statistics

The second shallow parsing extractor we used was the CASS parser (Abney, 1996), which uses cascaded finite state transducers to produce a limited depth parse of POS tagged text. We used the output of the Naïve Bayes POS tagger output as input to the CASS. The context relations used were extracted directly by the `tuples` program (using `e8` demo grammar) included in the CASS distribution. The FST parsing algorithm is very efficient and so CASS also ran very quickly. The times reported below include the Naïve Bayes POS tagging time.

The final, most sophisticated extractor used was the MINIPAR parser (Lin, 1998a), which is a broad-coverage principle-based parser. The context relations used were extracted directly from the full parse tree. Although fast for a full parser, MINIPAR was no match for the simpler extractors.

For this experiment we needed a large quantity of text which we could group into a range of corpus sizes. We combined the BNC and Reuters corpus to produce a 300 million word corpus. The respective sizes of each are shown in Table 2. The sentences were randomly shuffled together to produce a single homogeneous corpus. This corpus was split into two 150M word corpora over which the main experimental results are averaged. We then created smaller corpora of size $\frac{1}{2}$ down to $\frac{1}{64}$ th of each 150M corpus. The next section describes the method of evaluating each thesaurus created by the combination of a given context extraction system and corpus size.

5 Evaluation

For the purposes of evaluation, we selected 70 single word noun terms for thesaurus extraction. To avoid sample bias, the words were randomly selected from WordNet such that they covered a range of values for the following word properties:

occurrence frequency based on frequency counts from the Penn Treebank, BNC and Reuters;

number of senses based on the number of WordNet synsets and Macquarie Thesaurus entries;

generality/specificity based on depth of the term in the WordNet hierarchy;

abstractness/concreteness based on even distribution across all WordNet subtrees.

Table 3 shows some of the selected terms with frequency and synonym set data. For each term we extracted a thesaurus entry with 200 potential synonyms and their weighted Jaccard scores.

The most difficult aspect of thesaurus extraction is evaluating the quality of the result. The simplest method of evaluation is direct comparison of the extracted thesaurus with a manually created gold standard (Grefenstette, 1994). However on smaller corpora direct matching alone is often too coarse-grained and thesaurus coverage is a problem.

Our experiments use a combination of three thesauri available in electronic form: The Macquarie Thesaurus (Bernard, 1990), Roget's Thesaurus (Roget, 1911), and the Moby Thesaurus (Ward, 1996). Each thesaurus is structured differently: Roget's and Macquarie are topic ordered and the Moby thesaurus is head term ordered. Roget's is quite dated and has low coverage, and contains a deep hierarchy (depth up to seven) with terms grouped in 8696 small synonym sets at the leaves of the hierarchy. The Macquarie consists of 812 large topics (often in antonym related pairs), each of which is separated into 21174 small synonym sets. Roget's and the Macquarie provide sense distinctions by placing terms in multiple synonym sets. The Moby thesaurus consists of 30259 head terms and large synonym lists which conflate all the head term senses. The extracted thesaurus does not distinguish between different head senses. Therefore, we convert the Roget's and Macquarie thesaurus into head term ordered format by combining each small sense set that the head term appears in.

We create a gold standard thesaurus containing the union of the synonym lists from each thesaurus, giving a total of 23207 synonyms for the 70 terms. With these gold standard resources in place, it is possible to use precision and recall measures to calculate the performance of the thesaurus extraction systems. To help overcome the problems of coarse-grained direct comparisons we use three different types of measure to evaluate thesaurus quality:

1. Direct Match (DIRECT)

2. Precision of the n top ranked synonyms ($P(n)$)

3. Inverse Rank (INVR)

A match is an extracted synonym that appears in the corresponding gold standard synonym list. The direct match score is the number of such matches for each term. Precision of the top n is the percentage of matches in the top n extracted synonyms. In these experiments, we calculate this for $n = 1, 5,$ and 10 . The inverse rank score is the sum of the inverse rank of each match. For example, if matching synonyms appear in the extracted synonym list at ranks 3, 5 and 28, then the inverse rank score is $\frac{1}{3} + \frac{1}{5} + \frac{1}{28} = 0.569$. The maximum inverse rank score is 5.878 for a synonym list of 200 terms. Inverse rank is a good measure of subtle differences in ranked results. Each measure is averaged over the extracted synonym lists for all 70 thesaurus terms.

6 Results

Since MINIPAR performs morphological analysis on the context relations we have added an existing morphological analyser (Minnen et al., 2000) to the other extractors. Table 4 shows the improvement gained by morphological analysis of the attributes and relations for the SEXTANT 150M corpus.

The improvement in results is quite significant, as is the reduction in the representation space and number of unique context relations. The reduction in the number of terms is a result of coalescing the plural nouns with their corresponding singular nouns, which also reduces data sparseness problems. The remainder of the results use morphological analysis of both the words and attributes.

Table 5 summarises the average results of applying all of the extraction systems to the two 150M word corpora. The first thing to note is the time spent extracting contextual information: MINIPAR takes significantly longer to run than the other extractors. Secondly, SEXTANT and MINIPAR have quite similar results overall, but MINIPAR is slightly better across most measures. However, SEXTANT runs about 28 times faster than MINIPAR. Also, MINIPAR extracts many more terms and relations with a much larger representation than SEXTANT. This is partly because MINIPAR extracts more types of relations from the parse tree

Word	PTB Rank	PTB #	BNC #	Reuters #	Macquarie #	WordNet #	Min / Max	WordNet subtree roots
company	38	4076	52779	456580	8	9	3 / 6	entity, group, state
interest	138	919	37454	146043	12	12	3 / 8	abs., act, group, poss., state
problem	418	622	56361	63333	4	3	3 / 7	abs., psych., state
change	681	406	35641	55081	8	10	2 / 12	abs., act, entity, event, phenom.
house	896	223	47801	45651	10	12	3 / 6	act, entity, group
idea	1227	134	32754	13527	10	5	3 / 7	entity, psych.
opinion	1947	78	9122	16320	4	6	4 / 8	abs., act, psych.
radio	2278	59	9046	20913	2	3	6 / 8	entity
star	5130	29	8301	6586	11	7	4 / 8	abs., entity
knowledge	5197	19	14580	2813	3	1	1 / 1	psych.
pants	13264	5	429	282	3	2	6 / 9	entity
tightness	30817	1	119	2020	5	3	4 / 5	abs., state

Table 3: Examples of the 70 thesaurus evaluation terms with distribution information

Morph. Analysis	Space	Unique	Terms	DIRECT	P(1)	P(5)	P(10)	INVR
None	345Mb	14.70M	298k	20.33	32.5 %	36.9 %	33.6 %	1.37
Attributes	302Mb	13.17M	298k	20.65	32.0 %	37.6 %	32.5 %	1.36
Both	274Mb	12.08M	269k	23.74	64.5 %	47.0 %	39.0 %	1.86

Table 4: Effect of morphological analysis on SEXTANT thesaurus quality

than SEXTANT, and partly because it extracts extra multi-word terms. Amongst the simpler methods, $W(L_1R_1)$ and $W(L_{1,2})$ give reasonable results. The larger windows with low correlation between the thesaurus term and context, extract a massive context representation but the results are about 10% worse than the syntactic extractors.

Overall the precision and recall are relatively poor. Poor recall is partly due to the gold standard containing some plurals and multi-word terms which account for about 25% of the synonyms. These have been retained because the MINIPAR and CASS systems are capable of identifying (at least some) multi-word terms.

Given a fixed time period (of more than the four days MINIPAR takes) and a fixed 150M corpus we would probably still choose to use MINIPAR unless the representation was too big for our learning algorithm, since the thesaurus quality is slightly better.

Table 6 shows what happens to thesaurus quality as we decrease the size of the corpus to $\frac{1}{64}$ th of its original size (2.3M words) for SEXTANT. Halving the corpus results in a significant reduction for most of the measures. All five evaluation measures show the same log-linear dependence on the size of the corpus. Figure 1 shows the same trend for Inverse Rank evaluation of the MINIPAR thesaurus with a log-linear fitting the data points.

We can use the same curve fitting to estimate the-

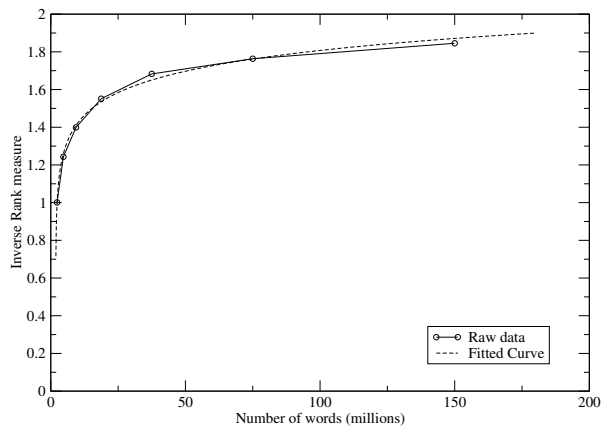


Figure 1: MINIPAR INVR scores versus corpus size

thesaurus quality on larger corpora for three of the best extractors: SEXTANT, MINIPAR and $W(L_1R_1)$. Figure 2 does this with the direct match evaluation. The estimate indicates that MINIPAR will continue to be the best performer on direct matching. We then plot the direct match scores for the 300M word corpus to see how accurate our predictions are. The SEXTANT system performs almost exactly as predicted and the other two slightly under-perform their predicted scores, thus the fitting is accurate enough to make reasonable predictions.

Figure 2 is a graph for making engineering decisions in conjunction with the data in Table 5. For instance, if we fix the total time and computational

System	Space	Relations	Unique	Terms	DIRECT	P(1)	P(5)	P(10)	INVR	Time
MINIPAR	399Mb	142.27M	16.62M	914k	24.55	61.5 %	46.5 %	40.5 %	1.85	4438.9m
SEXTANT	274Mb	53.07M	12.08M	269k	23.75	64.5 %	47.0 %	39.0 %	1.85	159.0m
CASS	186Mb	50.63M	9.09M	204k	20.20	48.5 %	38.5 %	32.5 %	1.51	173.7m
$W(L_1)$	117Mb	105.62M	7.04M	406k	20.60	51.5 %	40.0 %	32.5 %	1.56	6.8m
$W(L_{1,2})$	336Mb	206.02M	18.04M	440k	21.30	58.5 %	44.5 %	36.5 %	1.71	7.2m
$W(L_{1,2}^*)$	258Mb	206.02M	15.34M	440k	20.75	55.0 %	41.5 %	35.5 %	1.64	6.8m
$W(L_{1-3})$	570Mb	301.10M	30.62M	444k	20.50	60.0 %	43.5 %	37.0 %	1.69	8.2m
$W(L_{1-3}^*)$	388Mb	301.10M	22.86M	444k	19.85	48.5 %	39.5 %	33.5 %	1.53	8.2m
$W(L_1R_1)$	262Mb	211.24M	14.07M	435k	22.40	62.0 %	44.5 %	37.0 %	1.76	7.2m
$W(L_1R_1^*)$	211Mb	211.24M	12.56M	435k	20.90	54.5 %	42.5 %	34.5 %	1.64	7.2m

Table 5: Average thesaurus quality results for different extraction systems

Corpus	Space	Relations	Unique	Terms	DIRECT	P(1)	P(5)	P(10)	INVR
150.0M	274Mb	53.07M	12.08M	268.94k	23.75	64.5 %	47.0 %	39.0 %	1.85
75.0M	166Mb	26.54M	7.38M	181.73k	22.60	58.0 %	43.5 %	36.0 %	1.73
37.5M	98Mb	13.27M	4.36M	120.48k	21.75	54.0 %	41.0 %	34.5 %	1.62
18.8M	56Mb	6.63M	2.54M	82.33k	20.45	47.0 %	36.5 %	31.0 %	1.46
9.4M	32Mb	3.32M	1.44M	55.55k	18.50	40.0 %	32.5 %	27.5 %	1.29
4.7M	18Mb	1.66M	0.82M	37.95k	16.65	34.0 %	29.5 %	23.5 %	1.13
2.3M	10Mb	0.83M	0.46M	25.97k	14.60	27.5 %	25.0 %	19.5 %	0.93

Table 6: Average SEXTANT thesaurus quality results for different corpus sizes

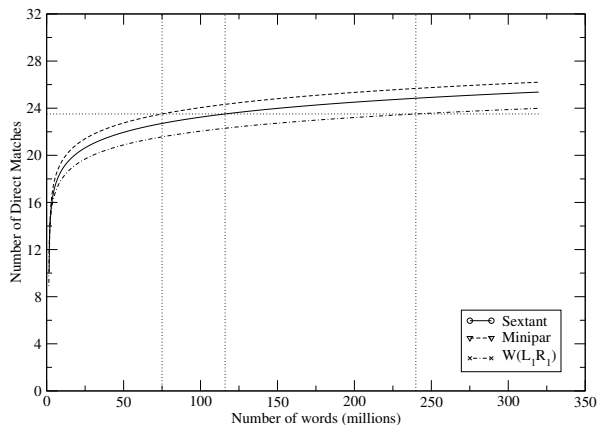


Figure 2: Direct matches versus corpus size

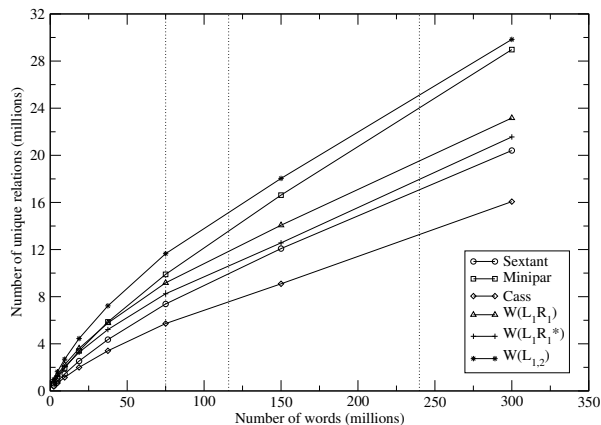


Figure 3: Representation size versus corpus size

resources at an arbitrary point, e.g. the point where MINIPAR can process 75M words, we get a best direct match score of 23.5. However, we can get the same resultant accuracy by using SEXTANT on a corpus of 116M words or $W(L_1R_1)$ on a corpus of 240M words. From Figure 5, extracting contexts from corpora of these sizes would take MINIPAR 37 hours, SEXTANT 2 hours and $W(L_1R_1)$ 12 minutes. Interpolation on Figure 3 predicts that the extraction would result in 10M unique relations from MINIPAR and SEXTANT and 19M from $W(L_1R_1)$. Figure 4 indicates that extraction would result in 550k

MINIPAR terms, 200k SEXTANT terms and 600k $W(L_1R_1)$ terms.

Given these values and the fact that the time complexity of most thesaurus extraction algorithms is at least linear in the number of unique relations and squared in the number of thesaurus terms, it seems SEXTANT may represent the best solution.

With these size issues in mind, we finally consider some methods to limit the size of the context representation. Table 7 shows the results of performing various kinds of filtering on the representation size. The FIXED and LEXICON filters run over the

System	Space	Relations	Unique	Terms	DIRECT	P(1)	P(5)	P(10)	INVR
SEXTANT 300M	431Mb	80.33M	20.41M	445k	25.30	61.0 %	47.0 %	39.0 %	1.87
SEXTANT 150M	274Mb	53.07M	12.08M	269k	23.75	64.5 %	47.0 %	39.0 %	1.85
SEXTANT FIXED	244Mb	61.17M	10.74M	265k	24.35	65.0 %	46.5 %	38.5 %	1.86
SEXTANT LEXICON	410Mb	78.69M	18.09M	264k	25.25	62.0 %	47.0 %	40.0 %	1.87
SEXTANT > 1	149Mb	67.97M	6.63M	171k	24.20	66.0 %	45.0 %	38.0 %	1.85
SEXTANT > 2	88Mb	62.57M	3.93M	109k	23.20	66.0 %	46.0 %	36.0 %	1.82

Table 7: Thesaurus quality with relation filtering

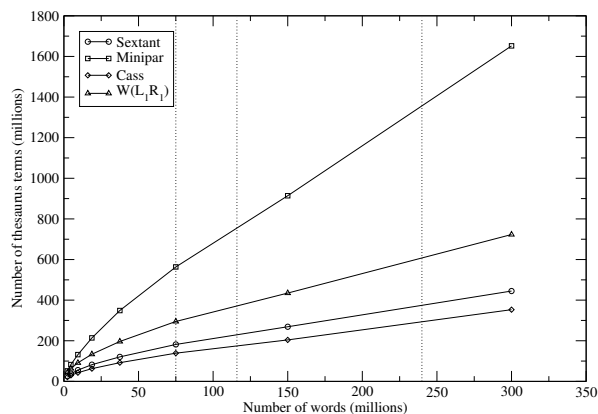


Figure 4: Thesaurus terms versus corpus size

full 300M word corpus, but have size limits based on the 150M word corpus. The FIXED filter does not allow any object/attribute pairs to be added that were not extracted from the 150M word corpus. The LEXICON filter does not allow any objects to be added that were not extracted from the 150M word corpus. The > 1 and > 2 filters prune relations with a frequency of less than or equal to one or two. The FIXED and LEXICON filters show that counting over larger corpora does produce marginally better results. The > 1 and > 2 filters show that the many relations that occur infrequently do not contribute significantly to the vector comparisons and hence don't impact on the final results, even though they dramatically increase the representation size.

7 Conclusion

It is a phenomenon common to many NLP tasks that the quality or accuracy of a system increases logarithmically with the size of the corpus. Banko and Brill, (2001) also found this trend for the task of confusion set disambiguation on corpora of up to one billion words. They demonstrated behaviour of different learning algorithms with very simple contexts on

extremely large corpora. We have demonstrated the behaviour of a simple learning algorithm on much more complicated contextual information on very large corpora.

Our experiments suggest that the existing methodology of evaluating systems on small corpora without reference to the execution time and representation size ignores important aspects of the evaluation of NLP tools.

These experiments show that efficiently implementing and optimising the NLP tools used for context extraction is of crucial importance since the increased corpus sizes make execution speed an important evaluation factor when deciding between different learning algorithms for different tasks and corpora. These results also motivate further research into improving the asymptotic complexity of the learning algorithms used in NLP systems. In the new paradigm, it could well be that far simpler but scalable learning algorithms significantly outperform existing systems.

Finally, the mass availability of online text resources should be taken on board. It is important that language engineers and computational linguists continue to try and find new unsupervised or (as Banko and Brill suggest) semi-supervised methods for tasks which currently rely on annotated data. It is also important to consider how information extracted by systems such as thesaurus extraction systems can be incorporated into tasks which use predominantly supervised techniques, e.g. in the form of class information for smoothing.

We would like to extend this analysis to at least one billion words for at least the most successful methods and try other tools and parsers for extracting the contextual information. However, to do this we must look at methods of compressing the vector-space model and approximating the full pair-wise comparison of thesaurus terms. We would also like

to investigate how this thesaurus information can be used to improve the accuracy or generality of other NLP tasks.

Acknowledgements

We would like to thank Miles Osborne, Stephen Clark, Tara Murphy, and the anonymous reviewers for their comments on drafts of this paper. This research is supported by a Commonwealth scholarship and a Sydney University Travelling scholarship.

References

- Steve Abney. 1996. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4):337–344, December.
- 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, 24–28 August.
- Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France, 9–11 July.
- 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.
- Sharon A. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 120–126, College Park, MD USA, 20–26 June.
- Stephen Clark and David Weir. 2001. Class-based probability estimation using a semantic hierarchy. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 95–102, Pittsburgh, PA USA, 2–7 June.
- Carolyn J. Crouch. 1988. Construction of a dynamic thesaurus and its use for associated information retrieval. In *Proceedings of the eleventh international conference on Research and Development in Information Retrieval*, pages 309–320, Grenoble, France, 13–15 June.
- James R. Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In *ACL-SIGLEX Workshop on Unsupervised Lexical Acquisition*, Philadelphia, PA USA. (to appear).
- Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Boston, USA.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th international conference on Computational Linguistics*, pages 539–545, Nantes, France, 23–28 July.
- Dekang Lin. 1998a. Dependency-based evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems, Proceedings of the First International Conference on Language Resources and Evaluation*, pages 234–241, Granada, Spain, 28–30 May.
- Dekang Lin. 1998b. An information-theoretic definition of similarity. In *Proceedings of the Fifteen International Conference on Machine Learning*, pages 296–304, Madison, WI USA, 24–27 July.
- Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust applied morphological generation. In *Proceedings of the First International Natural Language Generation Conference*, pages 201–208, Mitzpe Ramon, Israel, 12–16 June.
- Darren Pearce. 2001. Synonymy in collocation extraction. In *Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations, (NAACL 2001)*, pages 41–46, Pittsburgh, PA USA, 2–7 June.
- Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of the 31st annual meeting of the Association for Computational Linguistics*, pages 183–190, Columbus, Ohio USA, 22–26 June.
- Peter Roget. 1911. *Thesaurus of English words and phrases*. Longmans, Green and Co., London, UK.
- Gerda Ruge. 1997. Automatic detection of thesaurus relations for information retrieval applications. In *Foundations of Computer Science: Potential - Theory - Cognition, Lecture Notes in Computer Science*, volume LNCS 1337, pages 499–506. Springer Verlag, Berlin, Germany.
- Mark Sanderson and Bruce Croft. 1999. Deriving concept hierarchies from text. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 206–213, Berkeley, CA USA, 15–19 August.
- Grady Ward. 1996. *Moby Thesaurus*. Moby Project.