Experiments on Pattern-based Relation Learning

Willy Yap
NICTA Victoria Laboratory
Department of CSSE
University of Melbourne
willy@csse.unimelb.edu.au

Timothy Baldwin
NICTA Victoria Laboratory
Department of CSSE
University of Melbourne
tim@csse.unimelb.edu.au

Abstract

Relation extraction is a sub-task of Information Extraction (IE) that is concerned with extracting semantic relations—such as antonymy, synonymy or hypernymy—between word pairs from corpus data. Past work in relation extraction has concentrated on creating a small set of patterns that are good indicators of whether a word pair contains a semantic relation. In recent years, there has been work on using machine learning to automatically learn these patterns from text. We build on this research in running a series of experiments to investigate the impact of corpus type, corpus size and different parameter settings on learning a range of lexical relations.

1 Introduction

Information Extraction (IE) is the task of abstracting away from surface linguistic realisation in a text to expose its underlying informational context, usually relative to a predefined set of semantic predicates. Examples of typical IE tasks are fact discovery of corporate mergers and acquisitions from news articles (e.g. Company X was acquired by Company Y at time Z), and identifying that *animal* is a hypernym (= super-type) of *dog* from patterns of co-occurrence in a corpus. In this paper we focus on the latter task of relation extraction, over a range of binary lexical relations. That is, for a given word 2-tuple (x, y) and a lexical relation rel, we predict whether $rel(x, y)$ holds or not, based on analysis of co-occurrences of x and y in corpus data.

There are several motivations for relation extraction. The first is to provide knowledge (domain-specific or otherwise) for use in applications such as question answering or text summarisation, where relational data has been shown to enhance performance (Ravichandran and Hovy, 2002; Hovy and Lin, 1998). The second is to support the (semi-)automatic construction of semantic taxonomies, e.g. for specific domains or under-resourced languages (Grefenstette, 1994; Chklovski and Pantel, 2004; Snow et al., 2006). Semantic taxonomies are widely used across a range of Natural Language Processing tasks. Despite their usefulness, they tend to have low coverage due to the need for manual construction. With high-accuracy customisable relation extraction, we can hope to greatly reduce the manual overhead associated with constructing semantic taxonomies.

The main contribution of this work is to build on the work of Snow et al. (2005) on supervised relation extraction, over a larger range of relation types (hypernyms, synonyms and antonyms), exploring the impact of the choice of corpus, size of training data, and various parameter settings on extraction performance. For a given noun pair and relation type, the method performs automatic analysis of the patterns of sentential co-occurrence of the given nouns, and learns a classifier based on a set of training instances. Our results over the three lexical relations are some of the best achieved to date.

2 Related Work

There are two main approaches to relation extraction (Pantel et al., 2004). The first is the pattern-

based approach. Systems utilising this approach identify useful "lexico-syntactic patterns" that are strong indicators of a given lexical relation between a pair of words (Hearst, 1992). An example of a useful lexico-syntactic pattern is X, *such as* Y which strongly indicates that X is a hypernym of Y. After all instances of a predefined set of patterns are generated (either manually or automatically), the system then counts the number of times a given word pair occurs across those patterns. This is the basic approach that we adopt on in this paper, with the key difference that we do not use a predefined set of high-precision patterns, instead relying on our classifier to identify relevant patterns for a given lexical relation, which provide both positive and negative (strong and weak) evidence for a given noun pair belonging to that lexical relation.

The second approach is based on co-occurrence feature vectors. First, co-occurrence feature vectors representing the context of usage of each word are generated. Next, a clustering algorithm is deployed over the feature vectors to group words that have similar co-occurrence properties. Finally, clusters are labelled in some way. For example, there is a good chance that a system employing this approach would group *dog*, *cat*, *mouse*, and *bird* together in a single cluster. If the cluster were then labelled as *animal*, it could be inferred that *bird* is a hyponym of *animal* (Pantel and Ravichandran, 2004).

The pattern-based approach was popularised by Hearst (1992), in demonstrating the ability of a small set of lexico-syntactic patterns to extract hypernym noun pairs with high precision. Inspired by this work, there has been a great deal of work across different relation types (Girju et al., 2003; Yamada et al., 2007). However, there are two major drawbacks with this approach. Firstly, the patterns are expensive to generate – they require manual labor and the pattern designer requires a strong knowledge of the domain. Secondly, it is not always the case that a set of reliable patterns can be arrived at for a given relation type (e.g. the noisy set of patterns for identifying the telic role in Yamada et al. (2007)). Finally, while the patterns generally have high precision, they tend to suffer from low recall. This is due to the method being unable to detect pairs of words that correspond to the relation in question but are not expressed in one of the identified patterns.

Recent work has focused on iteratively bootstrapping these patterns (Stevenson and Greenwood, 2005; Xu et al., 2007) and automatically identifying them from text (Snow et al., 2005). The goal of such systems is to reduce the amount of manual effort and domain-specific knowledge required to identify or construct such patterns, and more importantly, to generate more patterns to improve the coverage of the system.

Snow et al. (2005) proposed a system that automatically identifies lexico-syntactic patterns indicating hypernym relations between a pair of nouns. They first parsed a corpora of 6 million sentences and in each sentence identified every noun pair. From this collection of noun pairs, they consulted WordNet to identify known hypernym pairs and known non-hypernym pairs, respectively. All lexico-syntactic patterns between noun pairs occurring in these two sets were then collected, and patterns with low frequency removed from the data. Finally, a feature vector is generated for each noun pair based on its occurrence across the post-filtered set of patterns, and a hypernym classifier is trained over all the known hypernym pairs and a fixed ratio of known non-hypernym pairs.

The work of both Stevenson and Greenwood (2005) and Snow et al. (2005) exemplifies *supervised* relation extraction, in the sense of requiring hand-labelled data or seed examples for a given relation type to initiate the learning process. If a new relation type is to be targeted, or a new domain explored (e.g. extracting BOOK–AUTHOR relation, or extracting holonym noun pairs), effort is required to source relevant training instances in sufficient quantities. In an attempt to relax this requirement, there has been increasing growth in *semi-supervised* IE, which leverages a handful of seed instances and large amounts of unannotated data (Bunescu and Mooney, 2007; Xu et al., 2007).

There also exist *unsupervised* relation extraction systems. For example, KNOWITALL is a complex unsupervised relation extraction system that contains various methods such as pattern learning, subclass extraction, and list extraction (Etzioni et al., 2005). Another unsupervised IE system that relies on a completely different methodology is *On-Demand Information Extraction* (Sekine, 2006). The basic idea is similar to KNOWITALL, but it uses different

methods in recognising patterns and tagging named entities. Both systems, however, have the same goal: eliminating the need to specifically repurpose an existing IE system for each relation type.

Finally and more recently, there has been work on a new style of IE termed *OpenIE*. Unlike traditional IE systems that require a specific relation to be predefined, Banko et al. (2007) and Banko and Etzioni (2008) proposed OpenIE as a means of extracting as many word pairs as possible that are possibly associated with a definable relation type. The basic premise behind this approach is that there is sufficient commonality to the lexico-syntactic patterns associated with various relations that token-level noun pairs can be identified in a manner akin to unlexicalised parsing. Although we believe this is a highly promising avenue of research, it is very difficult to judge the performance of OpenIE systems because the set of relations is open-ended.

3 Methodology

Our proposed approach to relation extraction builds directly off the work of Snow et al. (2005). Specifically, we tackle the task of noun relation extraction over a range of relation types by implicitly learning patterns which are positively and negatively correlated with a given relation type in the form of a supervised classifier. To demonstrate the generalisability of the method, we experiment with three noun semantic relations: antonymy, synonymy and hypernymy. Further, to investigate the impact of the type and size of the corpus on the performance of the system, we experiment with two different corpora.

As discussed in Section 2, patterns play a vital role in the relation extraction task. We thus require some way of representing the different lexico-syntactic configurations in which noun pairs occur. Ultimately we are interested in exploring a wide range of possibilities of preprocessing and removing the assumption of a parser, but for the purposes of this paper we use a dependency parser to generate these patterns in the form of dependency paths. In this, we take the parse for each sentence, identify all of the nouns, and generate the shortest dependency path between each pairing of nouns. We collect together all instances of a given noun pair across all sentences in our corpus, and calculate how many times it occurs with different patterns. A subset of these noun pairs is then selected for annotation according to a given semantic relation.

Our next step is to build a classifier to learn which patterns are positively and negatively correlated with the relation of interest. Tackling this problem as a machine learning task, we treat the noun pairs as our instances and the patterns of co-occurrence for a given noun pair as its features. We represent the frequency of occurrence of a given noun pair with a particular pattern as binary overlapping threshold buckets features, with thresholds defined by a power series of degree 2, i.e. $\{1, 2, 4, 8, ..., \}$, up to the maximum frequency of occurrence for any noun pair for a given pattern.

We next select a subset of noun pairs which are known to occur with the given relation as positive instances, and a subset of noun pairs which are known *not* to occur with the relation, and use these to train our classifier. As this process will typically be carried out relative to a set of seed instances or semi-developed lexical resource, in practical applications we expect to have ready access to some number of positive instances. Negative instances are more of an issue, in terms of both distinguishing unannotated from known negative instances, and also determining the ideal ratio of positive to negative instances in terms of optimising classifier performance. In their original research, Snow et al. (2005) got around this issue by taking a random sample of noun pairs and hand-annotating them, to get a feel for the relative proportion of positive to negative instances. This is a luxury that we may not be able to afford, however, and clearly slows down the development cycle. As part of this research, therefore, we investigate the impact of differing ratios of negative:positive training instances on our classifier performance.

In their work, Snow et al. (2005) applied two filters: (1) they filtered out noun pairs that do not occur across at least 5 distinct patterns, and (2) they used only patterns that appeared across at least 5 distinct noun pairs. In our work, we investigate how great an influence these parameters have on classifier performance across different relation types.

4 Resources

In this section we outline the various resources used by our system.

4.1 Corpora

We use two different corpora in our experiment: (1) English Gigaword corpus (LDC2003T05),[1] containing around 84 million sentences; and (2) Wikipedia July 2008 XML dump,[2] containing roughly 38 million sentences after preprocessing. Note that Wikipedia is less than half the size of Gigaword.

4.2 MINIPAR

MINIPAR is a fast and efficient broad-coverage dependency parser for English (Lin, 2004). We use MINIPAR to produce a dependency graph for each sentence in the English Gigaword corpus. Every word in a sentence is POS-tagged and represented as a node in the dependency graph. Dependency relations between word pairs (w_1, w_2) are represented by directed edges of the form:

$$w_1\text{,POS}_{w_1}\texttt{:relation:}\text{POS}_{w_2}\text{:}w_2$$

Since we are only interested in nouns in this research, we first identify all nouns, and from this form the set of noun pairs. The pattern between a noun pair in our experiment is defined as the shortest path of four or less edges linking that noun pair. We generalise the patterns by removing w_1 and w_2. Furthermore, we follow Snow et al. (2005) in post-processing the output of MINIPAR to: (a) include "satellite links", and (b) distribute patterns across all noun members of a conjunction. We refer the reader to the original paper for a detailed explanation of these enhancements.

4.3 WordNet

WordNet is a semantic lexical database of English, in the form of a network of synsets (Fellbaum, 1998). Each synset is a sense-indexed group of synonyms. WordNet is linked by semantic relations including antonymy (between words in synsets), hypernymy/hyponymy (between synsets), and holonymy/meronymy (between synsets). We used WordNet 3.0 in our experiments,[3] which is made up of a total of 117,798 unique noun synsets.

We use WordNet primarily as the source of class labelling in our experiments. This is done by first identifying all noun entries that appear in the semantic concordance (SemCor) file of WordNet (Landes et al., 1998). We then exhaustively generate all (directed) pairings of the 12,003 unique nouns obtained. For all noun pairings, we use WordNet to identify whether the pairing is in an antonym, synonym, or hypernym relation based on the first sense of each noun. If this is found to be the case, we classify that pairing as a positive instance relative to the given relation. We classify a pairing as a negative instance iff the given relation does not hold between any of the senses (first or otherwise) of the two nouns. This leaves two sets of noun pairs, which we ignore in the experimentation described in this paper: those where one or both nouns do not occur in SemCor, and those where both nouns occur in SemCor but the given relation holds for a non-first sense.

For hypernyms, we compare the first senses of nouns recursively up the WordNet hypernym hierarchy (considering all possible ancestors). That is, it is possible that noun senses which are not in SemCor themselves intervene between the two nouns. For antonyms, the situation is much simpler: we follow only a single edge between the senses of each noun. For synonyms it is simpler again: we simple look for membership in a common synset.

To illustrate this process, consider detecting the existence of a hypernym relation for the noun pair (X, Y), where X is *dog* and Y is *animal*. We first look up the hypernym of the first sense of *dog* (= dog_1). If the returned synset(s) contain the first sense of the second noun (*animal*), we stop and label the noun pair as a positive instance. However, in our example, $domestic\_animal_1$ and $canine_2$) are the immediate hypernyms of dog_1 and therefore we need to keep looking up the hierarchy from the two hypernyms. We stop when either we eventually find a match (in which case we label that noun pair as positive) or we do not find a match after we exhaus-

[1] `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05`

[2] `http://download.wikimedia.org/backup-index.html`

[3] We use the `WordNet::QueryData` Perl module to query the database.

tively traverse up the hierarchy. In the latter case, we exhaustively apply the same process again on the other sense combinations, e.g. dog_1 and $animal_2$. If there exists a hypernym relation between the non-first senses of the nouns, we ignore these noun pairs (the second category of the ignored noun pairs set). If and only if there is no hypernym relation between these noun pairs for all their sense combinations, we label the pairing as negative.

4.4 BSVM

BSVM is an efficient support vector machine (SVM) toolkit (Hsu and Lin, 2002), and is the only machine learner we use in this paper.[4] We use BSVM with its default settings ($C = 1, \epsilon = 0.001$), except for the choice of kernel where a simple linear kernel was found to be vastly superior to other kernels in terms of classification performance (but slower to converge).

5 Experiment and Results

5.1 Experimental Setup

As discussed in Section 3, Snow et al. (2005) used three parameters in his original work. The first parameter is a threshold over the number of patterns a given noun pair occurs with: noun pairs are included iff they occur with at least n patterns. We denote this parameter by $|np| \geq n$, and test three settings ($|np| \geq \{5, 10, 20\}$) in an attempt to ascertain whether noun pairs which occur across less patterns are significantly harder to classifier (in which case we expect the classifier performance to increase as the threshold value increases).

The second parameter determines which patterns are to be used as features in our data set, and acts as a means of feature selection. Only patterns which occur across at least m noun pairs are used in classification, which we denote as $|pat| \geq m$. Here, we experiment with the following settings: $|pat| \geq \{5, 10, 20, 50\}$. As this value rises, the feature space is thinned out but also becomes less sparse (as a given pattern will occur for more noun pairs). Once again, we are interested to see what impact this has on classifier performance.

The final parameter is the ratio of the negative to positive instances in our data. The ratio $|\mathbf{N}|/|\mathbf{P}| = r$ denotes that there are r times as many negative as positive instances in our data (based on the post-filtered count of positive noun pairs). We always use all available positive instances when evaluating over a given parameter selection and relation type. The number of negative instances for a given relation depends on the $|\mathbf{N}|/|\mathbf{P}|$ threshold and the number of positive instances for that relation. If we have pos positive instances and the ratio $|\mathbf{N}|/|\mathbf{P}| = r$, we pick the top $r \times pos$ most frequently appearing negative pairs in the data. We performed experiments on $|\mathbf{N}|/|\mathbf{P}| = \{1, 10, 25, 50, 100\}$.

In the original work, the parameters are set to $|np| \geq 5$, $|pat| \geq 5$, and $|\mathbf{N}|/|\mathbf{P}| = 50$.

Having built the data sets for the different relations over different parameters configurations, we evaluated classifiers based on 10-fold stratified cross validation. All the performance statistics reported here are the average of those performance scores across the 10 folds. Throughout evaluation, we measure relation extraction performance in terms of precision, recall and F-score ($\beta = 1$).

5.2 Baseline and Benchmark

The baseline for our experiment is a simple rule-based system that classifies a noun pair as having a given relation iff that noun pair occurs at least once in any one of the hand-crafted patterns associated with that relation.[5] These patterns were gathered from the work of Hearst (1992) (hypernyms), Widdows and Dorow (2002) (synonyms), and Lin et al. (2003) (antonyms), and are listed in Table 1. The performance of the baseline system is listed in Table 2, across the two corpora.

We use the result from Snow et al. (2005) as our benchmark. Since they only evaluate their system on hypernym relation, we can only compare the results of the two systems for hypernyms. Their best system—using a logistic regression machine learning algorithm—performs at an F-score of 0.348 for $|np| \geq 5$, $|pat| \geq 5$, and $|\mathbf{N}|/|\mathbf{P}| = 50$.

[4] In practice we experimented with maximum entropy, logistic regression, complement naive Bayes and various other algorithms, but found SVMs to be vastly superior in terms of their classification performance.

[5] We experimented with different settings for the threshold value, but found the value of 1 to produce the best results overall.

Relation	Patterns
hypernym	X and other Y X or other Y Y such as X Such Y as X Y including X Y, especially X
synonym	X and Y X or Y
antonym	from X to Y either X or Y

Table 1: Patterns used by the baseline system (NB for hypernymy, Y is a hypernym of X)

Baseline	Gigaword			Wikipedia		
	R	P	F	R	P	F
hypernym	.538	.054	.098	.641	.072	.129
synonym	.900	.021	.041	.940	.019	.037
antonym	.557	.178	.270	.106	.104	.105

Table 2: Baseline performance using the hand-crafted patterns (R = recall, P = precision, F = F-score)

5.3 Results

In our experiments, our primary interest is in the following areas: (1) the performance across different relation types; (2) the performance relative to a standardised baseline; (3) the performance across different corpora types and sizes; and (4) the effect of the parameter settings on performance.

First, the overall results across the three lexical relations across the English Gigaword and Wikipedia corpora are presented in Table 3, relative to the corresponding baseline results from Table 2. In order to track the relative change in these values in a normalised manner, we calculate the ERR over the baseline, based on the following calculation:

$$\text{ERR} = \frac{score_{\text{classifier}} - score_{\text{baseline}}}{1 - score_{\text{baseline}}}$$

These numbers are presented in brackets underneath the value (precision, recall or F-score) they are calculated relative to the numbers in Table 3.

In all cases, the precision and F-score are both well above the baseline, but recall actually falls below baseline for synonyms in particularly, largely due to the overly-permissive baseline pattern (i.e. any pair of nouns which occurs in a coordinate structure is considered to be a synonym pair). Comparing the different lexical relations, hypernyms are harder to learn than synonyms or antonyms, largely because of the ancestor-based interpretation of hypernyms (meaning that *organism* is a hypernym of *aardvark*, for example) vs. the more conventional interpretation of the other two lexical relations. Comparing Gigaword and Wikipedia, we see the Gigaword is markedly superior as a source of training data for hypernym learning (esp. in terms of recall), but that otherwise there is relatively little separating the two resources (despite Wikipedia being less than half the size of Gigaword). We further investigate this effect in our next set of experiments.

Next, we generate learning curves for each of the Gigaword and Wikipedia corpora, separately for each of the lexical relations. We plot the change in recall and precision as we increase the amount of data made available to the relation extraction method (increasing in increments of 10 million sentences),

	Gigaword			Wikipedia		
	R	P	F	R	P	F
hypernym	.625 (+.188)	.713 (+.697)	.666 (+.630)	.329 (−.869)	.690 (+.666)	.445 (+.363)
synonym	.897 (-.030)	.890 (+.888)	.892 (+.887)	.893 (−.783)	.899 (+.897)	.895 (+.891)
antonym	.780 (+.503)	.907 (+.887)	.820 (+.753)	.862 (+.846)	.874 (+.859)	.861 (+.845)

Table 3: Performance for hypernym, synonym and antonym learning over Gigaword and Wikipedia (R = recall, P = precision, F = F-score; numbers in brackets are the ERR relative to the corresponding baseline)

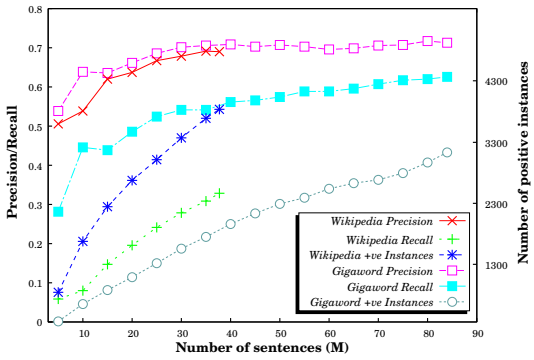

Figure 1: Learning curves for hypernym learning

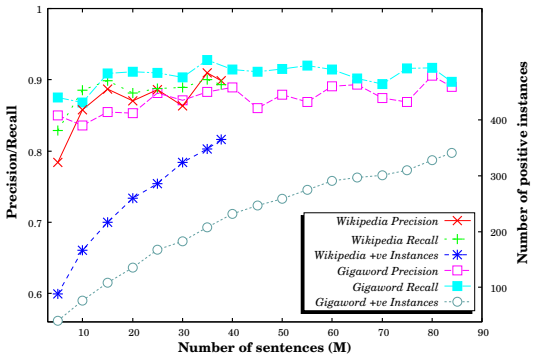

Figure 2: Learning curves for synonym learning

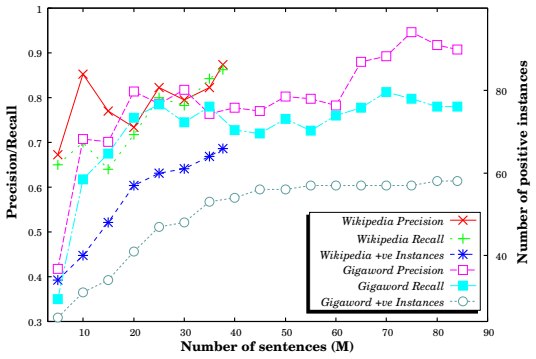

Figure 3: Learning curves for antonym learning

along with the number of positive instances that are observed in our dataset and form the basis of the evaluation. The curves are presented in Figures 1–3 for hypernyms, synonyms and antonyms, respectively.

We observe that the corpora have an interesting effect on the system performance. A consistent effect across all three lexical relations is that the number of positive instances observed in the data increases much more quickly for Wikipedia than Gigaword, because of its greater domain coverage and hence higher heterogeneity. In this sense, our original results have to be taken with a grain of salt: while we directly compare the recall, precision and F-score for the two corpora, the number of positive instances they are evaluated over (and implicitly the number of negative instances, based on the $|\mathbf{N}|/|\mathbf{P}|$ ratio value) is not consistent. As such, Wikipedia leads to a larger set of predictions with comparable precision, recall and F-score for synonyms and antonyms. Closer analysis of the results over hypernyms reveals that the recall is lower because the system is having to classify a larger set of noun pairs, including a higher proportion of lower-frequency, hard-to-classify pairs. Direct comparison is thus not fair, and further research is required to ascertain how the method is performing over noun pairs of different types.

Finally, looking at the effects of the system parameters, first with $|\mathbf{N}|/|\mathbf{P}|$ (the ratio of negative to positive instances), we vary the ratio over a wider range of $|\mathbf{N}|/|\mathbf{P}| = \{1, 10, 25, 50, 100\}$, and present the results in Table 4. Naturally, we expect there to be an overall drop in the numbers as the ratio increases, as the negative instances progressively overshadow the positive instances.

Unlike the effects of changing the values of the other two parameters, the effect of changing the value of the ratio parameter is substantial, with recall being particularly hard hit as the ratio increases. In fact, the recall actually drops below that of the baseline for $|\mathbf{N}|/|\mathbf{P}| = 100$ over hypernyms and synonyms, although the F-score is still comfortably above the baseline. This situation can be explained by the fact that the output is increasingly biased towards the negative instances. The performance of hypernyms suffers the most out of all three relations. This can be explained by the large number of hypernym positive instances that we have in our gold standard compared to the number of positive pairs for the other two relations (see Figure 1), meaning that the raw number of negative instances swamps the classifier more noticeably.

While we omit the full results from the paper, we also experimented with different settings for $|np|$ and $|pat|$, and observed that they tended not to affect the system performance. We also observed that the (marginally) best performance was achieved with the settings of $|np| \geq 5$ and $|pat| \geq 5$, which is the same settings used by Snow et al. (2005).

6 Discussion and Future Work

As seen in the ERR figures in Table 3, our systems outperformed the baseline in terms of F-score in all cases, across all relations.

With the same parameter settings, our system outperformed the system of Snow et al. (2005) at hypernym extraction (0.654 (Gigaword) and 0.445 (Wikipedia) vs. 0.348 F-score). However, this is not

$\|pat\| \geq 5$ $\|np\| \geq 5$	$\|\mathbf{N}\|/\|\mathbf{P}\| = 1$			$\|\mathbf{N}\|/\|\mathbf{P}\| = 10$			$\|\mathbf{N}\|/\|\mathbf{P}\| = 25$			$\|\mathbf{N}\|/\|\mathbf{P}\| = 50$			$\|\mathbf{N}\|/\|\mathbf{P}\| = 100$		
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
hypernym	.988 (+.974)	.986 (+.968)	.987 (+.971)	.917 (+.820)	.885 (+.864)	.901 (+.870)	.817 (+.604)	.772 (+.751)	.794 (+.759)	.625 (+.188)	.713 (+.697)	.666 (+.630)	.194 (−.745)	.425 (+.403)	.266 (+.212)
synonym	1.000 (+1.000)	.982 (+.964)	.991 (+.975)	.971 (+.710)	.941 (+.935)	.955 (+.946)	.945 (+.450)	.904 (+.900)	.921 (+.914)	.897 (-.030)	.890 (+.888)	.892 (+.887)	.850 (−.500)	.778 (+.776)	.809 (+.805)
antonym	.983 (+.962)	.973 (+.667)	.977 (+.925)	.936 (+.856)	.879 (+.737)	.903 (+.785)	.871 (+.709)	.902 (+.858)	.874 (+.790)	.780 (+.503)	.907 (+.887)	.820 (+.753)	.674 (+.264)	.843 (+.825)	.730 (+.672)

Table 4: Performance over different $|\mathbf{N}|/|\mathbf{P}|$ ratio values (R = recall, P = precision, F = F-score; numbers in brackets are the ERR relative to the corresponding baseline)

a strictly fair comparison as we used almost 15 times (Gigaword) and 7 times (Wikipedia) the amount of the data as they used in their experiment, with different machine learning algorithms, and the membership of positive and negative instances differed due to us enforcing the requirement that both nouns occur in SemCor. It is not possible to directly compare the results for synonyms and antonyms with other work, but again the results appear highly competitive with comparable research (e.g. Turney (2008)).

In future work, we plan to investigate the effect of including noun pairs where both nouns occur in SemCor but the given relation holds over a second or lower sense for one or more of the nouns (a class which is currently excluded from evaluation somewhat artificially). We noticed that there are quite a number of noun pairs that possess this relation when we build our gold standard. Also, we plan to perform the experiment on other relations, such as meronymy, as well as going beyond nouns and WordNet to look at learning qualia roles, for example.

7 Conclusion

We experimented with a great number of experiment settings for the pattern-based relation learning from corpus data. Building on the method of Snow et al. (2005), we have established that the method can be applied successfully across different semantic relations, and gained insights into the effects of different parameterisations on classifier performance. The experiments produced a very encouraging results, and suggested a number of promising directions for future research.

Acknowledgments

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*, pages 28–36, Columbus, Ohio.

Michele Banko, Michael Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pages 2670–2676, Hyderabad, India.

Razvan C. Bunescu and Raymond J. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of ACL*, pages 576–583, Prague, Czech Republic.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of EMNLP*, pages 33–40, Barcelona, Spain.

Oren Etzioni, Michael Cafarella, Doug Downey, Ann-Maria Popesu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

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

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2003. Learning semantic constraints for the automatic

discovery of part-whole relations. In *Proceedings of the HLT-NAACL*, pages 80–87, Edmonton, Canada.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING*, pages 539–545, Nantes, France.

Eduard H. Hovy and Chin-Yew Lin. 1998. Automated text summarization in SUMMARIST. In M. Maybury and I. Mani, editors, *Advances in Automatic Text Summarization*. MIT Press, Cambridge, USA.

Chih-Wei Hsu and Chih-Jen Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

Shari Landes, Claudia Leacock, and Randes Tengi. 1998. Building semantic concordances. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003)*, pages 1492–1493, Acapulco, Mexico.

Dekang Lin. 2004. Dependency-based evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain.

Patrick Pantel and Deepak Ravichandran. 2004. Automatically labelling semantic classes. In *Proceedings of the HLT-NAACL*, pages 321–328, Boston, USA.

Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. 2004. Towards terascale knowledge acquisition. In *Proceedings of COLING*, pages 771–777, Geneva, Switzerland.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL*, pages 41–47, Philadelphia, USA.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 731–738, Sydney, Australia.

Rion Snow, Dan Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in NIPS 17*, pages 1297–1304, Vancouver, Canada.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of COLING/ACL 2006*, pages 801–8, Sydney, Australia.

Mark Stevenson and Mark A. Greenwood. 2005. A semantic approach to IE pattern induction. In *Proceedings of ACL*, pages 379–386, Ann Arbour, USA.

Peter Turney. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 905–912, Manchester, UK.

Dominic Widdows and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proceedings of COLING*, pages 1093–9, Taipei, Taiwan.

Feiyu Xu, Hans Uszkoreit, and Hong Li. 2007. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *Proceedings of ACL*, pages 584–591, Prague, Czech Republic.

Ichiro Yamada, Timothy Baldwin, Hideki Sumiyoshi, and Nobuyuki Yagi. 2007. Automatic acquisition of qualia structure from corpus data. *IEICE Transactions on Information and Systems*, E90-D(10):1534–41.