# Experiments on Pattern-based Relation Learning

Willy Yap and Timothy Baldwin
NICTA Victoria Laboratory
Department of Computer Science and Software Engineering
University of Melbourne
{willy,tim}@csse.unimelb.edu.au

## ABSTRACT

Relation extraction is the task of extracting semantic relations—such as synonymy or hypernymy—between word pairs from corpus data. Past work in relation extraction has concentrated on manually creating templates to use in directly extracting word pairs for a given semantic relation from corpus text. Recently, there has been a move towards using machine learning to automatically learn these patterns. We build on this research by running experiments investigating the impact of corpus type, corpus size and different parameter settings on learning a range of lexical relations.

**Categories and Subject Descriptors:** I.2.6 [Learning]: Knowledge acquisition

**General Terms:** Algorithms, Experimentation

**Keywords:** relation extraction, information extraction

## 1. INTRODUCTION

Information Extraction (IE) is the task of abstracting away from surface linguistic realisation in a text to expose its underlying informational content, 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 corpus co-occurrence. 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 corpus co-occurrences of $x$ and $y$.

There are several motivations for relation extraction. The first is to provide knowledge for use in applications such as question answering or text summarisation, where relational data can enhance performance. The second is to support the (semi-)automatic construction of semantic taxonomies, e.g. for specific domains or under-resourced languages [9]. 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 [8] 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.

This paper is a shortened version of [11], which we refer the reader to for full methodological and experiment details.

## 2. RELATED WORK

There are several approaches to relation extraction. One is to take co-occurrence feature vectors representing the context of usage of each word, and use clustering to induce sets of words with similar co-occurrence properties. Finally, the clusters are labelled in some way. For example, there is a good chance that *dog*, *cat*, and *bird* will be grouped in the same cluster. If the cluster were then labelled as *animal*, it could be inferred that *bird* is a hyponym of *animal* [7].

The other main approach is to use lexico-syntactic patterns, as popularised by [4] for hypernym learning. The patterns are selected to be strong indicators of a given lexical relation between a pair of words, e.g. *X, such as Y* which strongly indicates that $X$ is a hypernym of $Y$. We adopt this approach 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 them for a given relation; the learned patterns provide both positive and negative evidence for a given noun pair belonging to that relation. The major drawbacks with this approach are: (1) patterns are expensive to generate – they require manual labour and domain expertise; (2) not all relations can be identified by a set of reliable patterns; and (3) while the patterns generally have high precision, they tend to suffer from low recall. Recent research has attempted to address the drawbacks mentioned above by iteratively bootstrapping these patterns [10] or automatically identifying them from text [8].

The research mentioned above exemplifies *supervised* relation extraction, because it requires hand-labelled data or seed examples to learn a given relation type. If a new rela-

*CIKM'09,* November 2–6, 2009, Hong Kong, China.
Copyright 2009 ACM 978-1-60558-512-3/09/11 ...$10.00.

tion type is to be targeted or a new domain explored (e.g. extracting the BOOK–AUTHOR relation), 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 [2].

More recently, there has been work on a new style of IE termed *OpenIE* [1]. Unlike traditional IE systems that require a specific relation to be predefined, OpenIE offers a means of extracting word pairs that are associated with an unspecified relation type.

# 3. METHODOLOGY

Our proposed approach to relation extraction builds directly off the work of [8]. 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 research, [8] addressed 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, [8] applied two filters: (1) noun pairs had to occur across at least 5 distinct patterns; and (2) patterns had to appear 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

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

We use MINIPAR in our experiments to parse sentences from the corpus. MINIPAR is a fast and efficient broad-coverage dependency parser for English [6]. It produces a dependency graph for each parsed sentence. 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: $w_1$,$\text{POS}_{w_1}$:$\texttt{relation}$:$\text{POS}_{w_2}$:$w_2$.

Since we are only interested in nouns, we first identify all nouns, and from this form the set of noun pairs. The pattern between a noun pair in our experiments is defined as the path of length four or less edges linking that noun pair. We generalise the patterns by removing $w_1$ and $w_2$. Furthermore, we follow [8] in post-processing the output of MINIPAR to: (a) include “satellite links”; and (b) distribute patterns across all noun members of a conjunction.

WordNet v3.0 [3] is used as the source of instance labels in our experiments. This is done by first identifying all noun entries that appear in the semantic concordance (SemCor) file of WordNet. 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 (recursively up the WordNet hierarchy), 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: (1) those where one or both nouns do not occur in SemCor; and (2) those where both nouns occur in SemCor but the given relation holds for a non-first sense.

All of our experiments are based on the BSVM machine learner [5], with a linear kernel and default settings otherwise ($C = 1, \epsilon = 0.001$).

# 5. EXPERIMENT AND RESULTS

## 5.1 Experimental Setup

As discussed in Section 3, [8] used three parameters in their 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.

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

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

We denote this parameter by $|np| \geq n$, and test three settings ($|np| \geq \{5, 10, 20\}$). We expect noun pairs which occur across less patterns to be harder to classifier, and the classifier performance to thus 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).

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. Given *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 with $|\mathbf{N}|/|\mathbf{P}| = \{1, 10, 25, 50, 100\}$.

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

Evaluation in all cases is based on 10-fold stratified cross validation, and the performance statistics reported here are the average of the 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 experiments 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. These patterns are listed in Table 1.

We use the result from [8] 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 logistic regression algorithm) performs at an F-score of 0.348 for $|np| \geq 5$, $|pat| \geq 5$, and $|\mathbf{N}|/|\mathbf{P}| = 50$.

## 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.

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

**Table 2: 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)**

First, the overall results across the three lexical relations for Gigaword and Wikipedia are presented in 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 corresponding precision, recall or F-score value.

In all cases, the precision and F-score are both well above the baseline, but recall actually falls below baseline for synonyms in particular, 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 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.

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.

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

**Table 3: 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)**

Finally, we investigate the effects of the system parameters in Table 3. With $|\mathbf{N}|/|\mathbf{P}|$ (the ratio of negative to positive instances), 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 $|\mathbf{N}|/|\mathbf{P}|$ 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, meaning that the raw number of negative instances swamps the classifier more noticeably.

We 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 $|pat| \geq 5$, as used by [8]. For full details, see [11].

## 6. DISCUSSION AND FUTURE WORK

As seen in the ERR figures in Table 2, our system 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 [8] at hypernym extraction (0.654 (Gigaword) and 0.445 (Wikipedia) vs. 0.348 F-score). However, this is not 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.

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.

## 7. CONCLUSION

We experimented with a great number of experiment settings for the pattern-based relational learning from corpus data. Building on the method of [8], 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 highly 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.

## 8. REFERENCES

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

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

[3] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA, 1998.

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

[5] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

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

[7] P. Pantel and D. Ravichandran. Automatically labelling semantic classes. In *Proceedings of the HLT-NAACL*, pages 321–328, Boston, USA, 2004.

[8] R. Snow, D. Jurafsky, and A. Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in NIPS 17*, pages 1297–1304, Vancouver, Canada, 2005.

[9] R. Snow, D. Jurafsky, and A. Y. Ng. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of COLING/ACL 2006*, pages 801–8, Sydney, Australia, 2006.

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

[11] W. Yap and T. Baldwin. Experiments on pattern-based relation learning. Working Paper, University of Melbourne, 2009. `http://repository.unimelb.edu.au/10187/4755`.