To Use a Treebank or Not – Which Is Better for Hypernym Extraction?

Erik Tjong Kim Sang
University of Groningen
The Netherlands
E-mail: e.f.tjong.kim.sang@rug.nl

Abstract

We compare two processing methods for a single natural language processing task. One uses a treebank created with a full parser while the other restricts itself to lexical and part-of-speech information. We show that for the task under investigation, automatic extraction of hypernym-hyponym pairs from text, the former does not outperform the latter. We compare the output of the two approaches and look for an explanation for this unexpected result.

1 Introduction

In recent years there has been an increased availability of treebanks created by advanced parsing tools. These offer detailed syntactic relations between words and phrases, unlike text corpora of about fifteen years ago which only offered words and their syntactic classes. We expect material from treebanks to be very useful for various natural language processing tasks, more than for example text that has just been processed by a part-of-speech tagger. In order to validate our expectation, we set up an experiment in which we compared the performance of the two in a single task. Contrary to our expectation, the output obtained from the treebank material turned out to be less reliable than that of the tagged text.

This paper describes the comparison experiment. After this introduction, we introduce the target task, extraction of hypernymy information from text, and describe how the experiment was set up. Next, we present the results of the comparison and discuss possible reasons for the outcome. In the final section, we present some concluding remarks.

2 Methods and experiments

We examine a single task: extraction of hypernymy information from text. A hypernym of a word X is a word Y which both contains the meaning of X and is broader. For example, an *orange* is a *fruit*, thus the word *fruit* is a hypernym of the word *orange*. If Y is a hypernym of X then X is a hyponym of Y. So *orange* is a hyponym of *fruit*.

Information about related hypernyms and hyponyms can be found in lexical resources such as WordNet [1] and EuroWordNet [8]. However, these resources are incomplete and therefore it is interesting to look for additional hypernymy information. One method for obtaining such information is to look in text for context patterns that link related words [3]. A phrase like *Y such as X* often contains a good hypernym candidate for *X*, namely Y.

For our work, we have used an extraction technique proposed by Snow, Jurafsky and Ng [6][1]. They searched in a large text corpus for sentences containing related word pairs, recorded the contexts of these pairs and used the context information for finding new pairs of related words. For example, if the words *orange* and *fruit* are known to be related and if the corpus contains the phrase *oranges and other fruits* then the phrase *X and other Y* can be used for finding candidate hypernyms for the word *X*.

Collecting phrases from text is a matter of counting how often a phrase appears in a text, either with related words or with unrelated words. Lexical resources usually do not contain information about unrelated words. Like Snow et al., we assume that two words are unrelated if they are both present in the resource while they have not been defined as related [6]. Therefore we only consider evidence for context phrases that relate two known words.

We combine the evidence of different context patterns in order to determine if two words are related. For this purpose, we apply the machine learning method Bayesian Logistic Regression [2] (also used by [6]). However, any other machine learner can be used for this task. Evaluation is performed with 10-fold cross validation which means that we divide the available material in ten sections and use each section as test set with the other nine as training set. The performance is measured by counting how many hypernym-hyponym pairs are found and how many of them are correct. We combine these two counts in precision scores. Since we obtained various different test set sizes, we also registered the number of positive cases (targets) of each test set.

The main goal of the experiment was to compare information available in a treebank that was built with a full parser, with linguistic annotations obtained from

[1] Similar work has been done by Nichols et al. [4].

a part-of-speech tagger. We used Dutch NRC newspaper section from the Alpino Treebank which was created by the parser with the same name [7][2]. The treebank contains 5.7 million sentences with over 100 million tokens. Like in the work of Snow et al., we limited the maximum size of the context phrases to four dependency links. Additionally, a single word modifying one of the two target words could be added to the phrase. An example of a context phrase is *Y like(modifies Y) X(complements like)*. Rather than the words in the text, the lemmas of these words were used in the phrases. We only used head words of noun phrases as target words.

The second data source was obtained by processing the same newspaper corpus with a part-of-speech tagger and lemmatizer that were trained on the Dutch CGN corpus as described in our earlier work [5]. A basic filter was used for identifying noun phrases: Det* Adj* N+. Like with the treebank data, target phrases consisted of lemmas, and linked head words (the final word) of noun phrases. These phrases were restricted to three center words to which a single word that modified one of the target words could be added. An example of such a phrase is *Y like X the(modifies X)*. We considered all nonhead words of the noun phrases as possible modifiers.

3 Results and discussion

We ran two hypernym extraction processes on the Dutch newspaper corpus, one using the treebank material while the other used the output of the part-of-speech tagger. Context patterns were extracted using hypernym-hyponym pairs taken from the Dutch part of EuroWordNet [8]. These patterns were used to extract candidate hypernym-hyponym pairs from the corpus. The evidence for these pairs was combined with Bayesian Logistic Regression and the results were evaluated using ten-fold cross-validation.

The results of the experiments can be found in Table 1. Since the extraction process produced different numbers of suggestions for the two data sources, we have performed additional evaluations where the number of proposed hypernym candidates was increased or decreased to match the size corresponding with the other data set (by changing the acceptance threshold of the machine learner). We registered a significantly higher precision score for the tagged data (41.8–18.6). However, the differences were a lot smaller when we corrected the numbers for the data size. We only registered a significant difference between the precision scores for the tagged data in comparison with the reduced output of the treebank data (41.8–33.3, $p < 0.05$).

[2]The output of the parser has *not* been manually corrected. The accuracy of the parser is about 90% F score for labaled dependency relations.

Data source	Targets	Found	Correct	Precision
Tagged data	675±24	225±15	94±9	41.8±3.7%
(adjusted threshold)		905±29	183±14	20.2±1.3%
Treebank data	1027±32	905±28	168±13	18.6±1.3%
(adjusted threshold)		225±15	75±9	33.3±3.3%

Table 1: Hypernym extraction results for the NRC newspaper corpus: number of related word pairs in the test set, number of extracted pairs, number of correct pairs, precision score and the associated standard deviations. There are two result lines for each data source: one with the default acceptance threshold and one with an adapted threshold to obtain the same number of accepted word pairs as with the other data source.

The extraction process that used the treebank data failed to outperform the process that only had access to tagged data and even failed to reach the same precision levels. We were surprised about this fact. How could the treebank patterns be more inaccurate than the lexical patterns? In order to find an answer to this question, we examined the hypernym-hyponym pairs suggested by the two best individual patterns measured by $F_{\beta=1}$ rate. Both the treebank data and the tagging data had the pattern *X and other Y* as best individual pattern. The precision scores of the two patterns were similar (23–25) but the treebank pattern missed more pairs found by the lexical pattern than vice versa (38–34) and generated more additional incorrect pairs (142–120, see Table 2).

From the output of the two approaches, we examined the correct pairs that were proposed by the lexical pattern but were missed by the treebank pattern. These are the pairs which potentially could have caused the precision of the treebank pattern to be lower than it should have been. There were 38 of such pairs. After examining them, we found four different causes for the misclassifications. Three of the pairs were missed because of two different causes.

Three of the four problems had their origin in processes outside of the parser. Sixteen pairs (42%) were indentified by the treebank pattern but were omitted from the test data because there were insufficient other patterns to support them (the process requires a minimum of five different extraction patterns to support a new hypernym-hyponym pair). One pair (3%) was missed because of a parse tree feature that was missing from the extraction software. And eight pairs (21%) were missed because of lemmatizing problems.

Sixteen (42%) of the missed hypernym-hyponym pairs originated from parsing problems. A frequent structure in the text is *X of Y and other Z*. The lexical pattern would always combine *Y* with *Z* but the treebank pattern could propose the pair *X*–*Z*

	correct		incorrect	
	+tree	-tree	+tree	-tree
+lexical	52	38	143	120
-lexical	34	–	142	–

Table 2: Confusion matrix for the best-performing lexical pattern *X and other Y* and the corresponding treebank pattern (tree). The matrix shows how many word pairs were found (+) or missed (-) by each of the two types of extraction patterns and, additionally, how many of these word pairs were related (correct) and how many were not.

as well as *Y–Z*, depending on the underlying tree structure. Any of these two could be correct but the examples from our data seem to suggest that shorter distance relations (like *Y–Z*) are more likely than longer distance relations (*X–Z*). Any time when the treebank patterns suggest a longer distance relation, they are predicting a low probability event. This could be an explanation for the fact that the treebank patterns achieve lower precision scores than lexical patterns which only suggest the more likely shorter distance relations.

We will forward the treebank pattern problems to the authors of the treebank. Some of these are solvable, like the attachment of *such as* to numbers like in *a million animals such as geese*. However, others are caused by tasks which are hard for any language processing system, like for example prepositional phrase attachment. We do not expect all parsing problems to be solved soon.

4 Concluding remarks

We have compared the effects of data sources on a single linguistic task: the extraction of hypernymy information from text. We compared extraction patterns built from a treebank with patterns that were generated from part-of-speech tagger output. When applied to a Dutch newspaper corpus, we found that, contrary to our expectations, the treebank patterns did not outperform the lexical patterns and failed to reach the same precision scores. Inspection of the data suggested that cases missed by the treebank patterns were caused by parsing errors, some of which will be hard to avoid.

We believe that treebanks are useful resources for natural language processing tasks and that they should enable higher quality output than data annotated with more shallow tools than a full parser. We hope that this study will contribute to improving treebank quality and to the knowledge of how they can be applied for

supporting natural language processing tasks. In the future, we will remain working towards these goals.

References

[1] Christiane Fellbaum. *WordNet – An Electronic Lexical Database*. The MIT Press, 1998.

[2] Alexander Genkin, David D. Lewis, and David Madigan. *BBR: Bayesian Logistic Regression Software*. www.stat.rutgers.edu/˜ madigan/BBR/, 2005.

[3] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of ACL-92*. Newark, Delaware, USA, 1992.

[4] Eric Nichols, Francis Bond, Takaaki Tanaka, Sanae Fujita, and Daniel Flickinger. Robust ontology acquisition from multiple sources. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*. Sydney, 2006.

[5] Erik F. Tjong Kim Sang. *Generating Subtitles from Linguistically Annotated Text*. Internal report Atranos project, WP4-12, University of Antwerp, 2003.

[6] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *NIPS 2005*. Vancouver, Canada, 2005.

[7] Gertjan van Noord. At last parsing is now operational. In Piet Mertens, Cedrick Fairon, Anne Dister, and Patrick Watrin, editors, *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, 2006.

[8] Piek Vossen. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publisher, 1998.