Automatic Relation Extraction – Can Synonym Extraction Benefit from Antonym Knowledge?

Anna Lobanova, Jennifer Spenader, Tim van de Cruys, Tom van der Kleij, Erik Tjong Kim Sang
University of Groningen
{a.lobanova@| j.spenader@| t.van.de.cruys@| a.a.j.van.der.kleij@ai.| e.f.tjong.kim.sang@}rug.nl

Abstract

We use automatically extracted word pairs from one lexical relation to filter out incorrect pairs of another relation. Initial results for improving Dutch synonyms by filtering out antonyms show a small precision improvement.

1 Introduction

Automatic extraction of lexical relations is useful for improving the coverage of existing computational lexical resources. For example, representation of lexical knowledge in WordNet (Fellbaum, 1998) is based on the *synsets*, or sets of synonyms like (*rich, affluent, flush, loaded, moneyed, wealthy*). Being able to extract synonyms automatically would lead to a consistent way of improving and extending the representation of synonyms in wordnets across different languages. However, an important problem of current distribution-based methods of synonym extraction is that they produce noise. As Lin et al. (2003) point out, an automatically obtained list of the top-20 distributionally similar words of *adversary* includes not only synonyms like *opponent* and *antagonist* but also contrasted words like *supporter* and even antonyms like *ally*:

> ***adversary**: enemy, foe, ally, antagonist, opponent, rival, detractor, neighbor, supporter, competitor, partner, trading partner, accuser, terrorist, critic, Republican, advocate, skeptic, challenger* (Lin et al., 2003)

This is due to a similar distribution of antonyms and synonyms in text (Lucerto et al., 2004). Lin et al. (2003) suggested to perform a two-step relation extraction approach in which synonym extraction is followed by a step in which semantically incompatible word pairs are filtered out. A pair of words was considered semantically incompatible if it occurred in the two surface patterns *from X to Y* and *either X or Y*. The results of the combined approach were good but the authors did not evaluate the impact of the second step.

Our assumption is that the extra filtering step is useful for improving the quality of automatic relation extraction, in particular synonyms. The goal of this paper is to validate this assumption. We present an experiment with Dutch synonym extraction in which erroneously extracted antonyms are filtered out in a post-process. We show that the filtering step does indeed improve the quality of the first extraction step.

In the next section, we describe methods we used to extract antonyms and synonyms automatically. In section 3, we show how the results for the two relations can be combined and present the results of this approach, our conclusions are summarized in section 4.

2 Automatic extraction of antonyms and synonyms

In this section we describe our work on automatic extraction of antonyms and synonyms. We used two pattern-based approaches to extract antonyms. The first uses two manually selected text patterns (section 2.1). In the second approach, text patterns indicating an antonym relation were *learned* from a collection of texts using a small set of antonym pairs as seeds (section 2.2). Hearst (1992) was the first to use preselected lexico-syntactic patterns to automatically extract hypernym-hyponym pairs from text. Since then, text patterns have been used to extract different lexico-semantic relations, in most cases hyponyms and meronyms (Berland and Charniak, 1999; Pantel and Pennacchiotti, 2006; Tjong Kim Sang and Hofmann, 2007). Synonym extraction has instead focused on using distributional methods. We present our work on automatic extraction of synonyms in section 2.3.

2.1 Antonyms derived from chosen patterns

For the first experiment we chose two text patterns which we expected to contain antonym pairs frequently:

- *zowel X als Y* (*X as well as Y*), for example *zowel mannen als vrouwen* (*men as well as women*)
- *tussen X en Y* (*between X and Y*), for example *tussen goed en kwaad* (*between good and evil*)

We searched in the Twente News Corpus (300 million words) for these text patterns and selected all lower case nouns X and Y which appeared inside both patterns at least twice. The result was a list of 270 antonym candidates. These candidates have been assessed by five native speakers of Dutch. They had the choice of labeling a word pair either as antonym (e.g. *rich/poor*), synonym (e.g. *rich/wealthy*), co-hyponym (e.g. *cat* and *dog* are co-hyponyms of *animal*) or unrelated (none of the above relations). The results of this assessment can be found in Table 1. Percentages of word pairs which received the same label from all participants can be found in the row Unanimous. Percentages for word pairs that received the same label from three or more participants are displayed in the row Majority. 14 pairs (5%) did not receive a majority label.

The precision of the two patterns was not very high. 34% of the pairs were labeled as antonyms but 54% were assigned the label co-hyponym. The other two categories occurred rarely.

This approach shows that a small number of text patterns is already useful for extracting candidate antonym pairs. Incorporating more text patterns could lead to finding more good pairs. Manual selection of patterns is time-consuming. It is hard to think of all possible productive patterns. Indeed, some infrequent patterns might still provide a valuable contribution. Learning patterns from text and antonym pair examples is a fast and more objective alternative. We discuss this approach next.

	Antonyms	Synon.	Co-hyp.	Unrel.
Majority	34%	0%	54%	7%
Unanimous	22%	0%	16%	0%

Table 1: Human evaluation of the 270 pairs extracted by means of chosen patterns. Word pairs could be classified as antonyms, synonyms, co-hyponyms or unrelated. 14 pairs (5%) did not receive a majority label.

2.2 Antonyms derived from learned patterns

To extract patterns automatically we used two sets of seeds consisting of 6 and 18 well established antonyms. The algorithm we used was based on the approach of Ravichandran and Hovy (2002). All sentences containing one of the seed pairs were extracted from Dutch CLEF corpus (Jijkoun et al., 2003), the antonyms were replaced by a wildcard token, 50 most frequently occurring patterns that contained seed pairs at least twice were used to find all word pairs that co-occurred in the positions of the wildcard tokens in the corpus. Depending on the number of times a pattern contained an already known antonym pair and the total number of times that pattern was found in the corpus, each pattern was given a score. Patterns with a score above the threshold were used to calculate the antonymy score (A_i) for each word pair that occurred in them. This score is the probability that the i-th pair is an actual antonym pair, given how often it occurred with each pattern (C_{i_j}) and the scores of these patterns (S_j):

$$A_i = 1 - \Pi_j (1 - S_j)^{C_{ij}} \qquad (1)$$

Pairs with a score $\geq$0.9 were used as new seeds in the following iteration. The entire process of identifying patterns and using those to extract new antonyms was repeated iteratively six times.

After six cycles, the seed sets of 6 and 18 elements had resulted in lists of respectively 1189 and 1355 antonym pairs. Pairs with a score $\geq$0.6 were checked by the human assessors. In the set of 6 seeds, 9 out of 197 checked pairs were antonyms according to EuroWordNet (5%). In the result set obtained with 18 seeds, 10 out of 172 checked pairs were antonyms according to EuroWordNet (6%). Pairs were then evaluated as antonyms, synonyms, co-hyponyms, or none of the above by five participants. The results are presented in Table 2.

The assessment results are comparable to those for the chosen text patterns in Table 1. The precision scores were around 30% but the number of extracted pairs was smaller (an average of 185 in comparison with the 270 of the chosen patterns). Note that the percentages of antonyms found by the assessors are a lot higher than the percentages in EuroWordNet. The antonym relation in EuroWordNet is incomplete.

	Antonyms	Synon.	Co-hyp.	Unrel.
6 seeds				
Majority	27%	1%	39%	31%
Unanimous	16%	0%	9%	15%
18 seeds				
Majority	33%	0%	35%	28%
Unanimous	20%	0%	9%	15%

Table 2: Human evaluation of the word pairs extracted by means of learned patterns: 197 with 6 seeds and 172 with 18 seeds. Word pairs could be classified as antonyms, synonyms, co-hyponyms or unrelated.

2.3 Automatic Extraction of Synonyms

The automatic extraction of synonyms has been carried out with standard dependency-based distributional similarity measures (Lin, 1998; Van de Cruys, 2006; Padó and Lapata, 2007). For each noun, a vector has been constructed, containing the frequencies of the dependency relations in which the noun appears. For example, a noun like *apple*, has features like red_{adj} and eat_{obj}. Dependency triples have been extracted from the CLEF corpus (Jijkoun et al., 2003). The 10,000 most frequent nouns have been used, together with the 60,000 most frequent dependency features, yielding a frequency matrix of 10K nouns by 60K dependency features. This matrix has been adapted with pointwise mutual information (Church and Hanks, 1990) for weighting purposes. Next, the noun by noun similarity matrix has been calculated using the cosine similarity measure. Finally, for each noun, all nouns that exceed a certain cosine similarity threshold are selected as the noun's candidate synonyms.

3 Using Antonyms in Synonym Extraction

We derived noun synonym candidates with distribution-based methods, cosine similarity and pointwise mutual information, as described in section 2.3. Next, we removed all synonym candidates which did not contain a word that was present in one of the two sets with antonyms derived in the previous experiments (see section 2; for the learned patterns, we used the set derived from 18 seeds). This resulted in two sets with unfiltered synonym candidates (114 and 80 word pairs, respectively) which will be used as baselines.

	cut-off (cosine)					
	.40	.30	.20	.18	.15	.10
Baseline (unfiltered)						
Precision	.008	.025	.053	.045	.036	.014
Recall	.003	.005	.035	.038	.048	.099
$F_{\beta=1}$	.004	.008	.042	.041	.041	.025
Filtered						
Precision	.008	.025	.055	.047	.039	.015
Recall	.003	.005	.035	.038	.048	.099
$F_{\beta=1}$	.004	.008	.042	.042	.043	.026

Table 3: Effects of filtering out antonyms derived with chosen patterns from a set of 114 candidate synonyms: a small positive effect on the low-cut-off sets.

Next, we removed from the synonym lists the candidate pairs that also occurred in the antonym lists. This produced two sets of filtered synonym pairs. We computed precision and recall scores for the filtered and the unfiltered synonym lists by comparing them with the synonyms in the Dutch part of EuroWordNet while using six different threshold values determined by the cosine similarity value of the word pairs. The results can be found in Tables 3 and 4.

When using antonyms derived with learned patterns, filtering out antonyms from a set of candidate synonyms had a large negative effect on the $F_{\beta=1}$ rates of high-cut-off sets (Table 4). The approach worked better with antonyms which had been extracted with chosen text patterns. Here we observed a small positive effect on the $F_{\beta=1}$ rates of low-cut-off sets (Table 3). The difference between the two approaches is surprising, given that the quality of the two sets of antonyms was similar according to human assessors (Tables 1 and 2).

Inspection of the results showed that the performance drop associated with the second set of antonyms was caused by a single synonym being present in the antonym list (see Table 5). If the synonym pair had not been classified as an antonym pair then the results of the second filter would have been similar to the first. This reveals a weakness of using learned patterns for identifying relations. The learner might use low-precision patterns which could be harmful for the quality of the results of the extraction process.

However, even without the incorrect pair in the antonym data, the positive effect would be small. In order to obtain a larger positive effect, we need more antonyms. This means that we either should use more data or use more extraction patterns.

	cut-off (cosine)					
	.40	.30	.20	.18	.15	.10
Baseline (unfiltered)						
Precision	.025	.035	.077	.097	.071	.024
Recall	.017	.025	.053	.091	.120	.174
$F_{\beta=1}$	.020	.029	.063	.094	.090	.042
Filtered						
Precision	.013	.023	.070	.090	.069	.024
Recall	.004	.013	.041	.078	.107	.161
$F_{\beta=1}$	.006	.017	.051	.084	.084	.042

Table 4: Effects of filtering out antonyms derived with learned patterns from a set of 80 candidate synonyms: a large negative effect on the high-cut-off sets.

CH	LE	
x		bewondering-afkeer (admiration-resent)
x		export-import (export-import)
x		jongen-meisje (boy-girl)
x	x	man-vrouw (man-woman)
	x	uitvoer-invoer (export-import)
	x	waarde-norm (value-norm)
	x	werkelijkheid-realiteit (reality-reality)
	x	werknemer-ambtenaar (employee-civil servant)
	x	werknemer-werkgever (employee-employer)

Table 5: The antonym pairs which were used for filtering: found by chosen patterns (CH) or by learned patterns (LE). Only one is a synonym: *werkelijkheid* - *realiteit* (both mean *reality*).

Antonym extraction was based on a text collection of 300 million words and it is unlikely that we will be able to collect a substantial number of extra text soon. Using more extraction patterns has the risk of generating additional false positives with a negative effect on the quality of antonyms.

4 Concluding remarks

We have described an experiment in which automatically extracted antonyms were used to filter out suspected errors from an automatically derived list of synonyms. We used two different methods for producing the antonyms and found that the ones produced by chosen high-quality text patterns were best suited for this approach. However, we only measured a small increase in the quality of the filtered synonym list in comparison with the unfiltered list.

In order to enlarge the observed positive effect, we need a larger set of antonyms. We have argued that both using more data and finding more extraction patterns will be difficult to achieve. One way to work around this problem is by replacing antonymy with a relation which is more frequent, for example, hypernymy. Future research will have to show if this will lead to improved relation extraction results.

References

Matthew Berland and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proceedings of ACL 99*. Maryland, MD, USA.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information & lexicography. *Computational Linguistics*, 16(1):22–29.

Christiane Fellbaum. 1998. *WordNet – An Electronic Lexical Database*. The MIT Press.

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

Valentin Jijkoun, Gilad Mishne, and Maarten de Rijke. 2003. Preprocessing documents to answer dutch questions. In *Proceedings of BNAIC'03*. Nijmegen, The Netherlands.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of IJCAI 2003*. Acapulco, Mexico.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of* COLING/ACL *98*, Montreal, Canada.

Cupertino Lucerto, David Pinto, and Héctor Jiménez-Salazar. 2004. An automatic method to identify antonymy relations. In *Workshop on Lexical Resources and the Web for Word Sense Disambiguation*. IBERAMIA 2004, Puebla, Mexico.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Patrick Pantel and Marco Pennacchiotti. 2006. Expresso: Leveraging generic patterns for utomatically harvesting semantic relations. In *Proceedings of ACL 2006*. Sydney, Australia.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL 2002*. Pittsburg, PA, USA.

Erik Tjong Kim Sang and Katja Hofmann. 2007. Automatic extraction of dutch hypernym-hyponym pairs. In *Proceedings of CLIN-2006*. Leuven, Belgium.

Tim Van de Cruys. 2006. Semantic clustering in dutch. In *Proceedings of CLIN-2005*, pages 17–32. University of Amsterdam, Amsterdam, The Netherlands.