Using a Treebank for Finding Opposites

Anna Lobanova

University of Groningen
Artificial Intelligence Dept.
E-mail: `a.lobanova@ai.rug.nl`

Gosse Bouma

University of Groningen
Information Science Dept.
E-mail: `g.bouma@rug.nl`

Erik Tjong Kim Sang

University of Groningen
Information Science Dept.
E-mail: `erikt@xs4all.nl`

Abstract

We present an automatic method for extraction of pairs of opposites (e.g. *hot-cold*, *top-bottom*, *buy-sell*) by means of dependency patterns that are learned from a 450 million word treebank containing texts from Dutch newspapers. Using small sets of seed pairs, we identify the best patterns for finding new pairs of opposites.

Treebanks are useful for generating dependency patterns expressing relations between words that occur far away from each other, something which is more difficult with textual patterns. Furthermore, textual patterns tend to find opposites expressed by the most frequent part-of-speech (PoS) category, viz. nouns ([17]). We examine whether dependency patterns can also be used for finding pairs of opposites of less frequent PoS classes: adjectives and verbs.

We successfully employ dependency patterns for extracting opposites but find that the best acquired patterns are too general and extract a lot of noise. We conclude that while syntactic information helps to identify opposites for less frequently co-occurring PoS categories, more data, e.g. available from the Web, should be used to improve the results.

1 Introduction

Recent years have produced increased efforts in research on automatic extraction of semantic relations like hyponymy, meronymy and synonymy. Yet, other relations, in particular, antonymy, have received little attention. In this paper, we present an automatic method for finding opposites by means of dependency patterns that are automatically acquired from a treebank of Dutch. We define opposites as a general class of antonyms that includes word pairs like *dead-alive*, *tall-short*, as well as incompatibles like *summer-winter*, *day-night*, *ask-answer*, etc. Our goal is to examine whether syntactic information is beneficial for identifying opposite words of different part-of-speech (PoS) categories.

Automatic extraction of opposites is useful for many NLP applications including sentiment analysis (e.g. by establishing the strength of antonymy of identified pairs [20]), automatic identification of contrastive relationships (see [19], [26]), and augmentation and verification of existing lexical resources, especially for languages other than English. A list of automatically found opposites can also be applied as a filter to improve the performance of automatic techniques for synonym, hyponym and meronym extraction ([18]), where antonym noise is a notorious problem ([16]).

Similarly to other lexically related words, opposites tend to co-occur with each other sententially and often they co-occur in so-called textual patterns like "*difference between X and Y*" or "*X as well as Y*" ([13]). However, opposites also occur outside of frequently used short textual patterns. For instance, in a Dutch example below (1), opposites *houden van - haten* ("to love" - "to hate") occur in parallel constructions outside of the scope of meaningful reoccurring textual patterns:

(1) Men houdt van Felicia, de Oprah Winfrey wannabe van Zuid-Afrika, of men haat haar.
People love Felicia, the Oprah Winfrey wannabe from South-Africa, or people hate her. (NRC, Dec 20, 2000)

Such cases are not rare. In fact, [13], who analysed 3000 newspaper sentences with well-established opposites, reports that in 38% of the sentences, opposites occurred outside of reasonable textual patterns. Because of this, many good instances can be missed, which in turn has a negative effect the recall of the pair extraction process. Dependency patterns can provide a plausible solution for this problem as they are acquired from treebank data, which contain syntactic relations between elements of a sentence and allow abstracting away from the surface structure. The dependency pattern *Verb1:conj* ← *of* → *conj:Verb2*, for example, links the two verbs in (1), representing the shortest path between them in the dependency tree.

While an increasing number of available treebanks allows to use syntactic information in relation extraction, using dependency patterns for finding opposites has not yet been done. Overall, there is no consensus as to whether such methods outperform techniques based on textual patterns. For example, [27] compared two automatic methods for hyponym-hypernym extraction for Dutch. In one method they used dependency patterns, while the other method relied on textual patterns which contained PoS category information. They found that both methods performed equally well. Results in an earlier study of [25], however, showed that dependency patterns outperformed textual patterns with PoS information for hypernym-hyponym extraction in English.

An important difference between antonymy as opposed to meronymy and hyponymy is that only antonymy relation can occur between words of more than one PoS category, including nouns, adjectives, and verbs. Exploring whether dependency patterns can find opposites that belong to different PoS categories is useful for understanding the benefits of syntactic information for relation extraction, as well as the extent to which dependency patterns differ from textual patterns. Since verb candidates are more likely to co-occur in a sentence further away from each other than nouns and adjectives, a method based on dependency patterns might be more productive for extraction of antonymous verbs rather than nouns and adjectives.

Alternatively, the antonym detection process might not be affected by the PoS categories of the candidates. Previous pattern-based work on extraction of opposites used textual patterns identified by means of adjective-adjective seeds ([17]). Interestingly, the majority of pairs they found were noun-noun pairs. Thus, by using dependency patterns with seeds that belong to several PoS categories, we can study whether syntactic information is more useful for pairs and relations that belong to certain PoS categories.

Outline The remainder of this article is organized as follows. In Section 2 we give an overview of previous pattern-based studies on relation extraction as well as existing work on antonym extraction. Our method is discussed in Section 3. The results are presented in Section 4. Our main finding is that dependency patterns are rather general and find not only opposites but also other frequently co-occurring pairs. The system performed best with adjective-adjective seeds, followed by nouns and verbs. The results are discussed and summarized in Section 5 where we also discuss directions of future work.

2 Previous work

A pattern-based method for relation extraction was originally proposed by [10] who suggested that patterns, in which words co-occur, signal lexical semantic relationships between them and, therefore, can be used to identify those relations. Using six manually identified textual patterns like *such NP as NP*, she found phrases, e.g. '*such authors as Shakespeare*' and used them to successfully extract facts like e.g. *Shakespeare* is a kind of *author*. In the 8.6 million word corpus of encyclopedic texts, Hearst found 153 candidate hyponym pairs, of which only 61 were listed in a hyponym relationship in WordNet [8], suggesting that the method could easily add useful relations missing in WordNet. As future work, Hearst suggested that a similar approach can be used to identify other lexical relationships.

Testing Hearst's suggestion, [1] used patterns to find meronyms from a newspaper corpus of 100 million words. Starting with a set of selected meronym pairs as seeds, they extracted all sentences that contained them and manually identified plausible patterns. The best two patterns were then enlisted to extract new pairs. They report an accuracy of 55% for the top 50 meronyms derived for six seeds based on the majority vote of the evaluation of the pairs by five human annotators.

Neither [10], nor [1] identified patterns automatically. Using a minimally supervised bootstrapping algorithm *Espresso*, [22] identified generic patterns automatically and used them to extract a range of relations including meronymy and hyponymy. Also beginning with seed pairs, they extracted all sentences these pairs co-occurred in in a 6.3 million word newspaper corpus and used those sentences to generalize patterns. All patterns were automatically evaluated based on pointwise mutual information ([4]). Top-10 best patterns were used to find new pairs. Extracted pairs were also evaluated using an association score between a given pair and a highly reliable pattern. Since by nature generic patterns are frequent and contain a lot of noise, pattern recall was increased by using the Web to retrieve more instances. Their method had high precision and also high recall. The obtained precision scores for the sample of 50 extracted instances of hyponyms and 50 extracted instances of meronyms with their top algorithm were between 73% and 85% (based on evaluation by two human annotators). Our algorithm is based on Espresso, but instead of textual patterns, we apply dependency patterns. As Pantel and Pennacchiotti mention themselves (2006: 3), the way patterns are defined and extracted does not affect the algorithm.

[25] were the first to use syntactic information to automatically derive dependency patterns to find hyponym-hypernym pairs in English. In their approach, they compared performance of a number of classifiers that as their features used noun-noun pairs extracted from a fully parsed six million word corpus of newspaper texts and different types of patterns, including dependency patterns and textual patterns. Their best logistic regression classifier was based on dependency patterns. It outperformed a classifier based on manually crafted patterns from [10]. According to the authors, their results indicate that dependency patterns are not only useful for identification of hyponymy relation but that they are better at hypernym-hyponym extraction than methods based on textual patterns.

The extent to which syntactic information is beneficial, is still disputed. In particular, [27] replicated Snow et al.'s approach on Dutch and compared it with a method based on textual patterns with PoS information. No significant differences were found between these methods. The largest effect was found for Wikipedia texts, where dependency patterns found 23% more related pairs than textual patterns. The authors argue that this affect can be overcome by adding 43% extra data.

Studies described above dealt with noun-noun pairs only. In this study we aim at finding a relation expressed not only by noun-noun but also adjective-adjective and verb-verb pairs. Using Espresso-based algorithm for finding meronyms, [12] conducted a detailed evaluation of the role seed types can play in extracting the target relation. They found that the best results were achieved using seeds that belonged to the same PoS class rather than mixed types. By using seeds for each PoS category, we examine how grammatical category of seeds affects generation of patterns and, consequently, the range of opposites found. It might be that a pattern-based method performs better with seeds of a certain PoS category, e.g. the most frequent one expressed by nouns, something that is addressed in our study.

Existing work on automatic extraction of opposites is based on surface patterns that do not capture any syntactic information. Starting with a small set of adjective-adjective seeds, [17] extracted all sentences that contained any seed pair from a newspaper corpus of Dutch (72 million words). Textual patterns were automatically constructed, and top-50 most frequently occurring patterns that contained one of seed pairs at least twice were used to find new instances of antonyms. Patterns consisted of five or more tokens as shorter patterns extracted too much noise. Both patterns and found instances were automatically scored. The scoring of patterns was based on how often they contained seed pairs and their overall frequency. The scoring of pairs was based on the number of times a pairs occurred with each pattern and its score. The algorithm was repeated iteratively six times, using pairs with scoring above a set threshold as new seeds at each iteration. All found pairs with scoring above 0.6 were evaluated by five human annotators. The results showed that surface patterns can be used to find not only a small range of well-established opposites but a wider class of pairs known as incompatibles. Still, the precision scores were considerably lower than those found with automatic hyponym and meronym extraction. Based on the majority vote by five annotators, for the set of six seeds they report a precision of 28% for pairs with scoring $\geq$0.6 when separating opposites from incompatibles (54 pairs), and a precision of 67% when opposites and incompatibles were treated as one group (129 pairs). The authors suggest that one of the reasons for the lower scores is that although all seeds were adjectives, most of found pairs consisted of nouns. Antonymy as a relation is best understood for adjectives whereas with nouns there is a unclear boundary between incompatibles like *summer-winter* and correlates like *suspect-witness*. This made evaluation of the results more difficult. Importantly, all correlates they found indicated some kind of contrast (e.g. a found pair *suspect-witness* as opposed to correlates *table-chair*) suggesting that their results could be useful for automatic identification of contrast relations.

The study conducted by [17] is similar to the *Espresso* method ([12]), but the ranking of patterns and pairs is based on a different metric, making it difficult to compare results. In this study, we present an *Espresso*-like algorithm that is using the same metric as [12].

3 Current Study

3.1 Materials

Corpus. We used a 450 million word version of Twente Nieuws Corpus of Dutch (TwNC, [21]) that consisted of 26 million sentences. The corpus consists of newswire texts from five daily Dutch newspapers.[1] The corpus was syntactically parsed by Alpino, a parsing system for Dutch aimed at parsing unrestricted texts ([28]). The parsing accuracy of Alpino is over 90% (tested on a set of 2256 newspaper

[1]Namely, *Algemeen Dagblad*, *NRC Handelsblad*, *Parool*, *Trouw* and *Volkskrant*.

Adjective-Adjective seeds	Noun-Noun seeds	Verb-Verb seeds
poor - rich	beginning - end	lose - win
open - closed	man - woman	give - take
large - small	day - night	buy - sell
fast - slow	question - answer	open - close
beautiful - ugly	advantage - disadvantage	find - lose
narrow - broad	peace - war	laugh - cry
dry - wet	top - bottom	end - begin
new - old	heaven - hell	increase - decrease
high - low	exit - entrance	save - spend
cold - hot	strength - weakness	confirm - deny
old - young	punishment - reward	succeed - fail
long - short	optimist - pessimist	ask - answer
happy - sad	husband - wife	attack - defend
active - passive	chaos - order	hate - love
right - wrong	predator - prey	fall - rise
dead - alive	employer - employee	exclude - include
heavy - light	fact - fiction	export - import
hard - soft	attack- defence	add - remove

Table 1: List of (translated) seed pairs for each part-of-speech category.

sentences ([28]), which is comparable to the state-of-the-art parsers for English ([5], [3], [15]).

Seeds. Seed sets were manually compiled from available lists of well-established opposites studied in psycholinguistic experiments (e.g. word association tests [7]) and corpus-based experiments (e.g. in terms of *breadth of co-occurrence* in [14]) and discussed in theoretical classifications ([6]). A preliminary study showed that these seeds outperformed seed sets that consisted of morphologically-related pairs (e.g. *known* - ***un**known*) or top-50 most frequent antonyms presented in the Dutch lexical database CORNETTO ([11]). A complete list of adjective-adjective, noun-noun and verb-verb seeds used in this study is presented in Table 1.

3.2 The Algorithm

Our method is based on the well-known minimally-supervised bootstrapping algorithm, *Espresso* ([22]). First, using seed pairs as tuples, dependency patterns that contained both words of a pair, were extracted from the treebank. Patterns that were found once were discarded. Next, patterns were automatically scored. The reliability of a pattern p, $r_\pi(p)$, given a set of input pairs I was calculated as its average strength of association across each input (seed) pair i in I, weighted by the reliability of each input pair i, $r_\iota(i)$:

$$r_\pi(P) = \frac{\sum_{i \in I} \left(\frac{pmi(i,p)}{max_{pmi}} * r_\iota(i) \right)}{|I|}$$

where $pmi(i, p)$ is the pointwise mutual information score (Church and Hanks 1990) between a pattern and an input pair, and $\max_{pmi}$ is the maximum pointwise mutual information between all patterns and all pairs. The reliability of initializing seed pairs was set to 1. Next, the top-k most reliable patterns were used to find new candidate pairs. We set the number of initial set of top patterns to 10, adding one extra pattern at each iteration.[2] The reliability of found pairs, $r_\iota(i)$ was estimated as follows:

$$r_\iota(i) = \frac{\sum_{p \in P} \left(\frac{pmi(i,p)}{max_{pmi}} * r_\pi(p) \right)}{|P|}$$

where P is the set of top-k found patterns.

The top-100 found pairs were used as new seeds in the next iteration. The process was repeated iteratively until at least 500 new pairs were acquired.

3.3 Evaluation

All found pairs were manually evaluated by three human annotators. They were asked to classify each pair as an opposite or a non-opposite. Opposites were described as words that belong to the same category but express the opposite of each other. We report a Fleiss's kappa score for inter-annotator's agreement (Randolph 2005). A score between 0.61 and 0.8 is considered to indicate a substantial agreement. In addition, we evaluated the results against CORNETTO, a newly available lexical resource for Dutch ([11]).[3] Finally, we calculated precision scores for each set of results, treating all pairs unanimously judged as opposites as *true positives*, pairs unanimously judged as non-opposites as *false positives* and discarding ambiguous pairs.

4 Results

4.1 Results for adjective-adjective pairs

Out of 519 pairs found with 18 adjective-adjective seeds, 34% (178 pairs) were judged as opposites by at least two annotators (82% of which received unanimous vote). They contained pairs like *automatisch - handmatig* ("automatic - manual"), *ziek - gezond* ("sick - healthy"), *leeg - vol* ("empty - full"). For 88% of those pairs (156) both words were listed in CORNETTO, but only 67 of them (43%) were linked

[2] Because we use a much bigger corpus than Pantel and Pennacchiotti [22], we do not retrieve additional instances of patterns from the web. We also do not use a discounting factor suggested in Pantel and Ravichandran (2004) and used in Pantel and Pennacchiotti [22] to control for the bias of *pmi* towards infrequent events. Instead we remove patterns and pairs that occur only once.

[3] This evaluation was done by means of a Python module PYCORNETTO developed by Erwin Marsi and available at http://code.google.com/p/pycornetto/.

Nr of iteration	Adjective-Adjective pairs	Noun-Noun pairs	Verb-Verb pairs
1	0.67	0.56	0.22
2	0.52	0.44	0.16
3	0.46	0.36	0.14
4	0.39	0.31	0.12
5	0.34	0.25	0.10

Table 2: Precision scores per iteration and PoS category (Adjective, Noun, Verb).

as opposites, indicating that for 57% of the valid pairs found by our method, the antonym relation was missing in the database. However, the majority of the candidate pairs, 66% (341 pairs), was unanimously judged as non-opposite. Among such pairs were e.g. *dood - zwaargewond* ("dead - heavily injured"), *politiek - zakelijk* ("political - bussinesslike"), *blij - tevreden* ("happy - contented") and others. Annotators achieved a Fleiss's kappa score of 0.73 indicating substantial agreement. Precision scores for each iteration, summarized in Table 2, show decreasing precision scores for later iterations. In particular, while precision score for the adjective seeds at the first iteration was 0.67, it decreased to 0.34 at the last iteration. One of the reasons for this can be that new pairs added at each following iteration make the results noisier. To investigate that we examined top-50 novel pairs extracted only at a given iteration. Among top-50 novel pairs found only at iteration one, 74% (37 pairs) were judged as opposites leading to a precision score of 0.86. At the last iteration only 26% of pairs (13) found only at that iteration were judged as opposites by the majority vote leading to a precision score of 0.26.

We also analysed the top patterns to see whether dependency patterns discovered by means of initial seeds were different from patterns discovered at later iterations with found seeds. The most frequent pattern at each iteration was *ANT1:conj ← or → conj:ANT2*, followed by patterns *ANT1:conj ← as well as → conj:ANT2* and *ANT1:conj ← neither nor → conj:ANT2*. Patterns found at first iteration were rather general and frequent, all ten of them were also found at each consequent iteration. Interestingly, our algorithm did not find an equivalent variant of one of the most frequent and productive textual patterns discovered with adjectival seeds by [17], namely *between X and Y*.

4.2 Results for noun-noun pairs

Out of 518 pairs found with 18 noun-noun seeds, 28% (143 pairs) were judged as opposites by at least two participants (72% of them received unanimous vote). Among pairs classified as opposites were pairs *kind - volwassene* ("child - grown up"), *tegenstander - vriend* ("adversary - friend"), *mislukking - succes* ("failure - success").

For 90% of pairs (128) that were judged as opposites, both words were listed in the CORNETTO database but only nine of them (7%) were linked as opposites. Thus, 93% of opposites are not captured by the lexical resource. Another 72%

(375 pairs) were judged by the majority vote as non-opposites. These pairs included many correlates, e.g. *politicus - sporter* ("politician - sportsmen"), *slip - top* ("underpants - top"), *fan - speler* ("fan - player"), as well as unrelated words like *rijkdom - vrede* ("wealth - peace") and *naam - talent* ("name - talent"). The annotators achieved a Fleiss's kappa score of 0.67 indicating sufficient agreement.

Again, as shown in Table 2, precision scores were higher at initial iterations (precision of 0.56 at iteration one), gradually decreasing 0.23 after iteration five. Analysis of top-50 novel pairs found at a given iteration showed that 58% of top-50 novel pairs at iteration one were judged as opposites leading to a precision score of 0.61. Only four novel pairs out of top-50 of the last iteration proved to be opposites. Thus the largest number of opposites were found at the first iteration.

Three most frequent patterns found with noun-noun pairs were general patterns *ANT1:conj* $\leftarrow$ *as well as* $\rightarrow$ *conj:ANT2*, *ANT1:conj* $\leftarrow$ *and* $\rightarrow$ *conj:ANT2* and *ANT1:conj* $\leftarrow$ *but* $\rightarrow$ *conj:ANT2*. A variant of pattern with connective *but* was found only in the third iteration with the set of adjective-adjective seeds. Unlike patterns with adjective-adjective seeds, half of patterns found with noun-noun seeds were longer and contained dependencies between subjects and objects.

4.3 Results for verb-verb pairs

The annotators agreed least on the classification of 518 pairs found with 18 verb-verb seeds, achieving a Fleiss's kappa score of 0.56. Contrary to our expectations, this set had the lowest precision scores out of the three PoS category sets. Namely, only 15% (78 pairs) were opposites according to the majority vote. They contained pairs like like *trouwen - scheiden* ("to marry - to divorce"), *verhoog - verminder* ("to raise - to decrease"), ontvangen - verzenden ("to receive - to send"). For 77 of them (99%), both words were found in CORNETTO but only 20 were marked as opposites, missing 74% of good instances. Among the 440 pairs judged as non-opposites were correlates *trouwen - samenwonen* ("to marry - to live together"), near-synonyms *bekijk - bezoek* ("to see over - to visit") and frequently co-occurring words like *downloaden - spelen* ("to download - to play").

Looking at the top-50 novel pairs found at a each given iteration only showed that the precision scores were very low at all five iterations ranging from 0.12 at iteration one to 0.04 at iteration five. This suggests that dependency patterns were not able to find many reliable instances of opposites neither with original seeds nor with seeds acquired during iterations.

Among the top three iteration patterns for verb-verb seeds were *to ANT1 or to ANT2*, *ANT1 or ANT2* and *be ANT1 or ANT2*. We also found variants of the patterns *ANT1 as well as ANT2*, *to ANT1 or to ANT2*, *neither ANT1 nor ANT2* and *ANT1 more than ANT2*. Thus, patterns found with each seed set were equivalent.

5 Discussion and Future Work

We have studied the application of dependency patterns learned from a treebank for the automatic identification of pairs of opposite words. We presented results for three PoS categories: adjective-adjective, noun-noun and verb-verb pairs. We showed that the results depended on the target PoS category. The best results were achieved for adjective pairs, followed by noun and verb pairs. Analysis of novel pairs found only at a given iteration showed that the most reliable pairs were found at the initial iterations (precision scores of 0.67 for adjectives, 0.56 for nouns and 0.22 for verbs). While results for top-50 novel adjective and noun antonym pairs are comparable with the results from similar pattern-based methods for finding meronyms ([12]) and hyponyms ([25]), contrary to our expectations, dependency patterns were not productive for finding opposites expressed by verbs. One of the reasons for this is that the best patterns found at each iteration are too general. Opposites expressed by verbs are also the least frequent category of sententially co-occurring pairs suggesting that this result might reflect the behavioural preferences of antonymous verbs rather than limitations of a particular automatic method.

Preference for short and general patterns is one of the main shortcomings of the present method. As a result, our algorithm is not able to discover an equivalent of one of the most productive textual patterns for finding opposites "*between X and Y*". Instead coordination construction "*X and Y*" is treated as the shortest path, dismissing the preposition *between*.

The lexical semantic relation of antonymy is not present in the most up-to-date available lexical resource for Dutch CORNETTO ([11]) for 57% of the correct opposites found with adjective seeds, 74% of the opposites found with verb seeds and 93% of the opposites found with noun seeds. This suggests that this method can be used as a supplementary means for improving existing databases. One way to improve the method itself would be to extend the algorithm so that it finds more instances with a given pattern by e.g., using Web data. However, given that Web provides immense data repository, it has yet to be determined whether we need to use dependency patterns or whether PoS tagging as a preprocessing step would be sufficient for antonym harvesting.

References

[1] Berland, M. and E. Charniak (1999) Finding Parts in Very Large Corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 57–64.

[2] Charles, W.G. and G. A. Miller (1989) Contexts of Antonymous Adjectives. *Applied Psycholinguistics* 10:3, pp. 357–375.

[3] Charniak, E. (2000) A Maximum-Entropy-Inspired Parser. In *Proceedings of*

the First Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2000), pp. 132–139.

[4] Church, K. W. and P. Hanks (1990) Word Association Norms, Mutual Information and Lexicography. *Computational Linguistics* 16:1, pp. 22–29.

[5] Collins, M. (1996) A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-1996)*, pp. 184–191.

[6] Cruse, A. (1986) *Lexical Semantics.* Cambridge: Cambridge University Press

[7] Deese J.E. (1964) The Associative Structure of Some Common English Adjectives. *Journal of Verbal Learning and verbal Behaviour* 3:5, pp. 347–357.

[8] Fellbaum Ch. (1998) *WordNet: An Electronic Lexical Database*. MIT Press.

[9] Randolph, J. J. (2005) *Free-marginal multirater kappa: An alternative to Fleiss' fixed-marginal multirater kappa.* Paper presented at the Joensuu University Learning and Instruction Symposium 2005, Joensuu, Finland.

[10] Hearst, M. (1992) Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pp. 539–545.

[11] Ales Horak and Piek Vossen and Adam Rambousek (2008) The Development of a Complex-structured Lexicon Based on WordNet. In *Proceedings of the 4th International Global WordNet Conference (GWC-2008)*, pp. 200–208.

[12] Ittoo, A. and G. Bouma (2010) On Learning Subtypes of the Part-Whole Relation: Do Not Mix Your Seeds. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, pp. 1328–1336.

[13] Jones S. (2002) *Antonymy: A Corpus-based Perspective*. London: Routlegde.

[14] Jones, S., Paradis C., Murphy M.L. and C. Willners (2007) Googling for Opposites - a Web-based Study of Antonym Canonicity. *Corpora* 2:2, pp. 129–155.

[15] Lin D. and Pantel, P. (2001) Induction of Semantic Classes from Natural Language Text. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, pp. 317-322, San Francisco, CA.

[16] Lin D., Zhao Sh., Qin L. and M. Zhou (2003) Identifying Synonyms among Distributionally Similar Words. In *Proceedings of IJCAI-2003*, pp. 1492–1493.

[17] Lobanova, A., van der Kleij, T. and J. Spenader (2010) Defining Antonymy: a Corpus-based Study of Opposites by Lexico-syntactic Patterns. *International Journal of Lexicography* 23:1, pp. 19–53.

[18] Lobanova, A., Spenader, J., van de Cruys, T., van der Kleij, T. and E. Tjong Kim Sang (2009) Automatic Relation Extraction - Can Synonym Extraction Benefit from Antonym Knowledge? In *Proceedings of WordNets and other Lexical Semantic Resources - between Lexical Semantics, Lexicography, Terminology and Formal Ontologies (NODALIDA2009 workshop)*, pp. 17–20.

[19] Marcu, D. and A. Echihabi (2002) An Unsupervised Approach to Recognizing Discourse Relations. in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)* pp. 7–12.

[20] Mohammad, S., Dorr, B. and G. Hirst (2008) Computing Word-Pair Antonymy. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 982–991.

[21] Ordelman R. J. F. (2002) *Twente Nieuws Corpus (TwNC)*. Parlevink Language Technology Group. University of Twente.

[22] Pantel, P. and M. Pennacchiotti (2006) Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics (COLING/ACL-06)*, pp. 113–120.

[23] Pantel, P. and D. Ravichandran. (2004) Automatically Labeling Semantic Classes. In *Proceedings of the HLT-NAACL conference*

[24] Ravichandran, D. and E. Hovy (2002) Learning Surface Text Patterns for a Question Answering System. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pp. 41–47.

[25] Snow, R., Jurafsky D. and Ng A. (2005) Learning Syntactic Patterns for Automatic Hypernym Discovery. In *Proceedings of the NIPS 17*, pp. 1297–1304.

[26] Spenader, J. and G. Stulp (2007) Antonymy in Contrast Relations. In *Seventh International Workshop on Computational Semantics*

[27] Tjong Kim Sang, E. and K. Hofmann (2009) Lexical Patterns or Dependency Patterns: Which Is Better for Hypernym Extraction? In *Proceedings of CoNLL-2009*, pp. 174–182.

[28] van Noord, G. (2006) At Last Parsing is Now Operational. In Mertens, P., Fairon C., Dister A. and P. Watrin (eds.) *TALN 2006. Verbum Ex Machina. Actes de la 13e Conference sur le Traitement Automatique des Langues Naturelles*, pp. 20–42.