Learning Antonyms with Paraphrases and a Morphology-aware Neural Network

Sneha Rajana* **Chris Callison-Burch*** **Marianna Apidianaki*Ψ** **Vered Shwartz$^{\Phi}$**
*Computer and Information Science Department, University of Pennsylvania, USA
ΨLIMSI, CNRS, Université Paris-Saclay, 91403 Orsay
$^{\phi}$Computer Science Department, Bar-Ilan University, Israel
{srajana,ccb,marapi}@seas.upenn.edu vered1986@gmail.com

Abstract

Recognizing and distinguishing antonyms from other types of semantic relations is an essential part of language understanding systems. In this paper, we present a novel method for deriving antonym pairs using paraphrase pairs containing negation markers. We further propose a neural network model, *AntNET*, that integrates morphological features indicative of antonymy into a path-based relation detection algorithm. We demonstrate that our model outperforms state-of-the-art models in distinguishing antonyms from other semantic relations and is capable of efficiently handling multi-word expressions.

1 Introduction

Identifying antonymy and expressions with contrasting meanings is valuable for NLP systems which go beyond recognizing semantic relatedness and require to identify specific semantic relations. While manually created semantic taxonomies, like WordNet (Fellbaum, 1998), define antonymy relations between some word pairs that native speakers consider antonyms, they have limited coverage. Further, as each term of an antonymous pair can have many semantically close terms, the contrasting word pairs far outnumber those that are commonly considered antonym pairs, and they remain unrecorded. Therefore, automated methods have been proposed to determine for a given term-pair (x, y), whether x and y are antonyms of each other, based on their occurrences in a large corpus.

Charles and Miller (1989) put forward the co-occurrence hypothesis that antonyms occur together in a sentence more often than chance. However, non-antonymous semantically related words such as hypernyms, holonyms, meronyms, and near-synonyms also tend to occur together more often than chance. Thus, separating antonyms from pairs linked by other relationships has proven to be difficult. Approaches to antonym detection have exploited distributional vector representations relying on the distributional hypothesis of semantic similarity (Harris, 1954; Firth, 1957) that words co-occurring in similar contexts tend to be semantically close. Two main information sources are used to recognize semantic relations: path-based and distributional. Path-based methods consider the *joint* occurrences of the two terms in a given sentence and use the dependency paths that connect the terms as features (Hearst, 1992; Roth and Schulte im Walde, 2014; Schwartz et al., 2015). For distinguishing antonyms from other relations, Lin et al. (2003) proposed to use antonym patterns (such as *either* X *or* Y and *from* X *to* Y). Distributional methods are based on the *disjoint* occurrences of each term and have recently become popular using word embeddings (Mikolov et al., 2013; Pennington et al., 2014) which provide a distributional representation for each term. Recently, combined path-based and distributional methods for relation detection have also been proposed (Shwartz et al., 2016; Nguyen et al., 2017). They showed that a good path representa-

Paraphrase Pair	Antonym Pair
not sufficient/insufficient	sufficient/insufficient
insignificant/negligible	significant/negligible
dishonest/lying	honest/lying
unusual/pretty strange	usual/pretty strange

Table 1: Examples of antonyms derived from PPDB paraphrases. The antonym pairs in column 2 were derived from the corresponding paraphrase pairs in column 1.

tion can provide substantial complementary information to the distributional signal for distinguishing between different semantic relations.

While antonymy applies to expressions that represent **contrasting** meanings, paraphrases are phrases expressing the **same** meaning, which usually occur in similar textual contexts (Barzilay and McKeown, 2001) or have common translations in other languages (Bannard and Callison-Burch, 2005). Specifically, if two words or phrases are paraphrases, they are unlikely to be antonyms of each other. Our first approach to antonym detection exploits this fact and uses paraphrases for detecting and generating antonyms (*The dementors* ***caught*** *Sirius Black/ Black could* ***not escape*** *the dementors*). We start by focusing on phrase pairs that are most salient for deriving antonyms. Our assumption is that phrases (or words) containing negating words (or prefixes) are more helpful for identifying opposing relationships between term-pairs. For example, from the paraphrase pair (caught/*not* escape), we can derive the antonym pair (caught/escape) by just removing the negating word 'not'.

Our second method is inspired by the recent success of deep learning methods for relation detection. Shwartz et al. (2016) proposed an integrated path-based and distributional model to improve hypernymy detection between term-pairs, and later extended it to classify multiple semantic relations (Shwartz and Dagan, 2016) (LexNET). Although LexNET was the best performing system in the semantic relation classification task of the CogALex 2016 shared task, the model performed poorly on synonyms and antonyms compared to other relations. The path-based component is weak in recognizing synonyms, which do not tend to co-occur, and the distributional information caused confusion between synonyms and antonyms, since both tend to occur in the same contexts. We propose *AntNET*, a novel extension of LexNET that integrates information about negating prefixes as a new morphological pattern feature and is able to distinguish antonyms from other semantic relations. In addition, we optimize the vector representations of dependency paths between the given term pair, encoded using a neural network, by replacing the embeddings of words with negating prefixes by the embeddings of the base, non-negated, forms of the words. For example, for the term pair *unhappy/joyful*, we record the negating prefix of ***un****happy* using a new path feature and replace the word embedding of *unhappy* with *happy* in the vector representation of the dependency path between *unhappy* and *sad*. The proposed model improves the path embeddings to better distinguish antonyms from other semantic relations and gets higher performance than prior path-based methods on this task. We used the antonym pairs extracted from the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013; Pavlick et al., 2015b) in the paraphrase-based method as training data for our neural network model.

The main contributions of this paper are:

- We present a novel technique of using paraphrases for antonym detection and successfully derive antonym pairs from paraphrases in the PPDB, the largest paraphrase resource currently available.

- We demonstrate improvements to an integrated path-based and distributional model, showing that our morphology-aware neural network model, AntNET, performs better than state-of-the-art methods for antonym detection.

2 Related Work

Paraphrase Extraction Methods Paraphrases are words or phrases expressing the same meaning. Paraphrase extraction methods that exploit distributional or translation similarity might however propose paraphrase pairs that are not meaning equivalent but linked by other types of relations. These methods often extract pairs having a related but not equivalent meaning, such as contradictory pairs. For instance, Lin and Pantel (2001) extracted 12 million "inference rules" from monolingual text by exploiting shared dependency contexts. Their method learns paraphrases that are truly meaning equivalent, but it just as readily learns contradictory pairs such as *(X rises, X falls)*. Ganitkevitch et al. (2013) extract over 150 million paraphrase rules from parallel corpora by pivoting through foreign translations. This multilingual paraphrasing method often learns hypernym/hyponym pairs, due to variation in the discourse structure of translations, and unrelated pairs due to misalignments or polysemy in the foreign language. Pavlick et al. (2015a) added interpretable semantics to PPDB (see Section 3.1 for

Method	#pairs
(x,y) from paraphrase (x̃,y)/(x,ỹ)	80,669
(x, paraphrase(y)), (paraphrase(x), y)	81,221
(x, synset(y)), (synset(x), y)	692,231

Table 2: Number of unique antonym pairs derived from PPDB at each step. Paraphrases and synsets were obtained from PPDB and WordNet respectively.

details) and showed that paraphrases in this resource represent a variety of relations other than equivalence, including contradictory pairs like *nobody/someone* and *close/open*.

Pattern-based Methods Pattern-based methods for inducing semantic relations between a pair of terms (x, y) consider the lexico-syntactic paths that connect the joint occurrences of x and y in a large corpus. A variety of approaches have been proposed that rely on patterns between terms in a corpus to distinguish antonyms from other relations. Lin et al. (2003) used translation information and lexico-syntactic patterns to extract distributionally similar words, and then filtered out words that appeared with the patterns 'from X to Y' or 'either X or Y' significantly often. The intuition behind this was that if two words X and Y appear in one of these patterns, they are unlikely to represent a synonymous pair. Roth and Schulte im Walde (2014) combined general lexico-syntactic patterns with discourse markers as indicators for the specific semantic relations between word pairs (e.g. contrast relations might indicate antonymy and elaborations may indicate synonymy or hyponymy). Unlike previous pattern-based methods which relied on the standard distribution of patterns, Schwartz et al. (2015) used patterns to learn word embeddings. They presented a symmetric pattern-based model for representing word vectors in which antonyms are assigned to dissimilar vector representations. More recently, Nguyen et al. (2017) presented a pattern-based neural network model that exploits lexico-syntactic patterns from syntactic parse trees for the task of distinguishing between antonyms and synonyms. They applied HypeNET Shwartz et al. (2016) to the task of distinguishing between synonyms and antonyms, replacing the direction feature with the distance in the path representation.

Source	#pairs
WordNet	18,306
PPDB	773,452

Table 3: Number of unique antonym pairs derived from different sources. The number of pairs obtained from PPDB far outnumbers the antonym pairs present in EVALution and WordNet.

3 Paraphrase-based Antonym Derivation

Existing semantic resources like WordNet (Fellbaum, 1998) contain a much smaller set of antonyms compared to other semantic relations (synonyms, hypernyms and meronyms). Our aim is to create a large resource of high quality antonym pairs using paraphrases.

3.1 The Paraphrase Database

The Paraphrase Database (PPDB) contains over 150 million paraphrase rules covering three paraphrase types: lexical (single word), phrasal (multiword), and syntactic restructuring rules, and is the largest collection of paraphrases currently available. PPDB . In this paper, we focus on lexical and phrasal paraphrases up to two words in length. We examine the relationships between phrase pairs in the PPDB focusing on phrase pairs that are most salient for deriving antonyms.

3.2 Antonym Derivation

Selection of Paraphrases We consider all phrase pairs from PPDB (p_1, p_2) up to two words in length such that one of the two phrases either begins with a negating word like *not*, or contains a negating prefix.[1] We chose these two types of paraphrase pairs since we believe them to be the most indicative of an antonymy relationship between the target words. There are 7,878 unordered phrase pairs of the form (p_1', p_2) where p_1' begins with 'not', and 183,159 phrases of the form (p_1', p_2) where p_1' contains a negating prefix.

Paraphrase Transformation For paraphrases containing a negating prefix, we perform morphological analysis to identify and remove the negating prefixes. For a phrase pair like *un*happy/sad, an antonymy relation is derived between the base form of the negated word, without the negation prefix, and its paraphrase (happy/sad). We use

[1] Negating prefixes include *de*, *un*, *in*, *anti*, *il*, *non*, *dis*

Unrelated	Paraphrases	Categories	Entailment	Other relation
much/worthless	correct/that's right	Japan/Korea	investing/ increased investment	twinkle/dark
disability/present	simply/merely	black/red	efficiency/ operational efficiency	naw/not gonna
equality/gap	till/until	Jan/Feb	valid/equally valid	access/available

Table 4: Examples of different types of non-antonyms derived from PPDB.

MORSEL (Lignos, 2010) to perform morphological analysis and identify negation markers. For multi-word phrases with a negating word, the negating word is simply dropped to obtain an antonym pair (e.g. different/*not* identical → different/identical). Some examples of PPDB paraphrase pairs and antonym pairs derived from them are shown in Table 1. The derived antonym pairs are further expanded by associating the synonyms (from WordNet) and lexical paraphrases (from PPDB) of each phrase with the other phrase in the derived pair. While expanding each phrase in the derived pair by its paraphrases, we filter out paraphrase pairs with a PPDB score (Pavlick et al., 2015a) of less than 2.5. In the above example, **un***happy/sad*, we first derive *happy/sad* as an antonym pair and expand it by considering all synonyms of *happy* as antonyms of *sad* (e.g. *joyful/sad*), and all synonyms of *sad* as antonyms of *happy* (e.g. *happy/gloomy*). Table 2 shows the number of pairs derived at each step using PPDB. In total, we were able to derive around 773K unique pairs from PPDB. This is a much larger dataset than existing resources like WordNet and EVALution as shown in Table 3.

Analysis We performed a manual evaluation of the quality of the extracted antonyms by randomly selecting 1000 pairs classified as 'antonym' and observed that the dataset contained about 63% antonyms. Errors mostly consisted of phrases and words that do not have an opposing meaning after the removal of the negation pattern. For example, the equivalent pair *till/until* that was derived from the PPDB paraphrase rule *not till/until*. Other non-antonyms derived from the above methods can be classified into unrelated pairs (background/figure), paraphrases or pairs that have an equivalent meaning (admissible/permissible), words that belong to a category (Africa/Asia), pairs that have an entailment relation (valid/equally valid) and pairs that are related but not with an antonym relationship (twinkle/dark). Table 4 gives some examples of categories of non-antonyms.

Annotation Since the pairs derived from PPDB seemed to contain a variety of relations in addition to antonyms, we crowdsourced the task of labelling a subset of these pairs in order to obtain the true labels.[2] We asked workers to choose between the labels: antonym, synonym (or paraphrase for multi-word expressions), unrelated, other, entailment, and category. We showed each pair to 3 workers, taking the majority label as truth.

4 LSTM-based Antonym Detection

In this section we describe AntNET, a long short term memory (LSTM) based, morphology-aware neural network model for antonym detection. We first focus on improving the neural embeddings of the path representation (Section 4.1), and then integrate distributional signals into our network resulting in a combined method (Section 4.2).

4.1 Path-based Network

Similarly to prior work, we represent each dependency path as a sequence of edges that leads from x to y in the dependency tree. We use the same path-based features proposed by Shwartz et al. (2016) for recognizing hypernym relations: lemma and part-of-speech (POS) tag of the source node, the dependency label, and the edge direction between two subsequent nodes. Additionally, we also add a new feature that indicates whether the source node is negated.

Rather than treating an entire dependency path as a single feature, we encode the sequence of edges using a long short term memory network (Hochreiter and Schmidhuber, 1997). The vectors obtained for the different paths of a given (x, y) pair are pooled, and the resulting vector is used for classification. The overall network structure is depicted in Figure 1.

[2] 5884 pairs were randomly chosen and were annotated on www.crowdflower.com

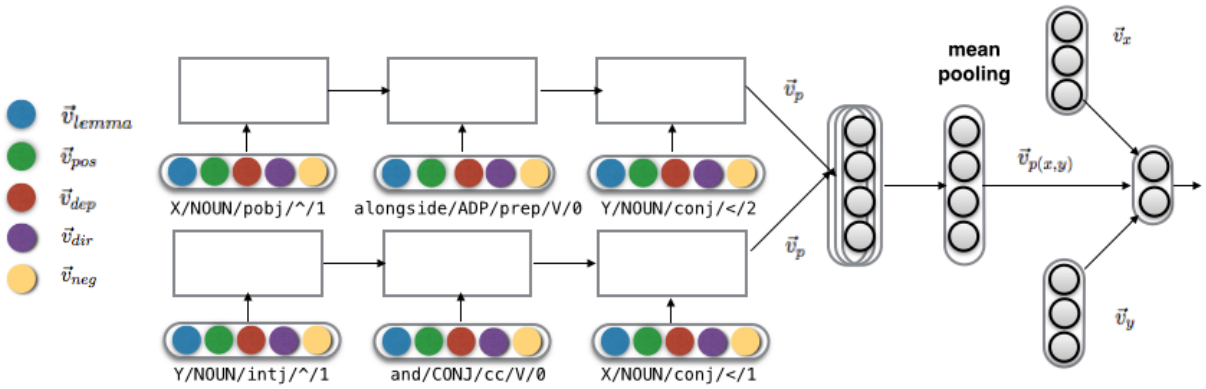

Figure 1: Illustration of the AntNET model. Each pair is represented by several paths and each path is a sequence of edges. An edge consists of five features: lemma, POS, dependency label, dependency direction, and negation marker.

Edge Representation We denote each edge as $lemma/pos/dep/dir/neg$. We are only interested in checking if x and/or y have negation markers but not the intermediate edges since negation information for intermediate lemmas is unlikely to contribute to identifying whether there is an antonym relationship between x and y. Hence, in our model, neg is represented in one of three ways: *negated* if x or y is negated, *not-negated* if x or y is not negated, and *unavailable* for the intermediate edges. If the source node is negated, we replace the lemma by the lemma of its base, non-negated, form. For example, if we identified *unhappy* as a 'negated' word, we replace the lemma embedding of *unhappy* by the embedding of *happy* in the path representation. The negation feature will help in separating antonyms from other semantic relations, especially those that are hard to distinguish from, like synonyms.

The replacement of a negated word's embedding by its base form's embedding is done for a few reasons. First, words and their polar antonyms are more likely to co-occur in sentences compared to words and their negated forms. For example, *Neither happy nor sad* is probably a more common phrase than *Neither happy nor unhappy*, so this technique will help our model to identify an opposing relationship between both types of pairs, *happy/unhappy* and *happy/sad*. Second, a common practice for creating word embeddings for multi-word expressions (MWEs) is by averaging over the embeddings of each word in the expression. Ideally, this is not a good representation for phrases like *not identical* since we lose out on the negating information obtained from *not*. Indicating the presence of *not* using a negation feature and replacing the embedding of *not identical* by *identical* will increase the classifier's probability of identifying *not identical/different* as paraphrases and *identical/different* as antonyms. And finally, this method helps us distinguish between terms that are seemingly negated but are not in reality (e.g. *invaluable*). We encode the sequence of edges using an LSTM network. The vectors obtained for all the paths connecting x and y are pooled and combined, and the resulting vector is used for classification. The vector representation of each edge is the concatenation of its feature vectors:

$$\vec{v}_{edge} = [\vec{v}_{lemma}, \vec{v}_{pos}, \vec{v}_{dep}, \vec{v}_{dir}, \vec{v}_{neg}]$$

where $\vec{v}_{lemma}, \vec{v}_{pos}, \vec{v}_{dep}, \vec{v}_{dir}, \vec{v}_{neg}$ represent the vector embeddings of the negation marker, lemma, POS tag, dependency label and dependency direction, respectively.

Path Representation The representation for a path p composed of a sequence of edges $edge_1, edge_2, .., edge_k$ is a sequence of edge vectors: $p = [\vec{edge}_1, \vec{edge}_2, ..., \vec{edge}_k]$. The edge vectors are fed in order to a recurrent neural network (RNN) with LSTM units, resulting in the encoded path vector $\vec{v}_p$.

Classification Task Given a lexical or phrasal pair (x, y) we induce patterns from a corpus where each pattern represents a lexico-syntactic path

connecting x and y. The vector representation for each term pair (x, y) is computed as the weighted average of its path vectors by applying average pooling as follows:

$$\vec{v}_{p(x,y)} = \frac{\sum_{p \in P(x,y)} f_p . \vec{v}_p}{\sum_{p \in P(x,y)} f_p} \quad (1)$$

$\vec{v}_{p(x,y)}$ refers to the vector of the pair (x, y); $P(x, y)$ is the multi-set of paths connecting x and y in the corpus and f_p is the frequency of p in $P(x, y)$. The vector $\vec{v}_{p(x,y)}$ is then fed into a neural network that outputs the class distribution c for each class (relation type), and the pair is assigned to the relation with the highest score r:

$$c = softmax(MLP(\vec{v}_{p(x,y)})) \quad (2a)$$

$$r = argmax_i c[i] \quad (2b)$$

MLP stands for Multi Layer Perceptron and can be computed with or without a hidden layer (equations 4 and 5 respectively).

$$\vec{h} = tanh(W_1 . \vec{v}_{p(x,y)} + b_1) \quad (3)$$

$$MLP(\vec{v}_{p(x,y)}) = W_2 . \vec{h} + b_2 \quad (4)$$

$$MLP(\vec{v}_{p(x,y)}) = W_1 . \vec{v}_{p(x,y)} + b_1 \quad (5)$$

W refers to a matrix of weights that projects information between two layers; b is a layer-specific vector of bias terms and $\vec{h}$ is the hidden layer.

4.2 Combined Path-based and Distributional Network

The path-based supervised model in Section 4.1 classifies each pair (x, y) based on the lexico-syntactic patterns that connect x and y in a corpus. Inspired by the improved performance of Shwartz et al.'s (2016) integrated path-based and distributional method over a simpler path-based algorithm, we integrate distributional features into our path-based network. We create a combined vector representation using both the syntactic path features and the co-occurrence distributional features of x and y for each pair (x, y). The combined vector representation for (x, y), $\vec{v}_{c(xy)}$, is computed by simply concatenating the word embeddings of x ($\vec{v}_x$) and y ($\vec{v}_y$) to the path-based feature vector $\vec{v}_{p(x,y)}$:

$$\vec{v}_{c(xy)} = [\vec{v}_x, \vec{v}_{p(x,y)}, \vec{v}_y] \quad (6)$$

5 Experiments

We experiment with the path-based and combined models for antonym identification by performing two types of classification: binary and multiclass classification.

Train	Test	Val	Total
5,122	1,829	367	7,318

Table 5: Number of instances present in the train/test/validation splits of the crowdsourced dataset.

5.1 Dataset

Neural networks require a large amount of training data. We use the labelled portion of the dataset that we created using PPDB, as described in Section 3. In order to induce paths for the pairs in the dataset, we identify sentences in the corpus that contain the pair and extract all patterns for the given pair. Pairs with an antonym relationship are considered as positive instances in both classification experiments. In the binary classification experiment, we consider all pairs related by other relations (entailment, synonymy, category, unrelated, other) as negative instances. We also perform a variant of the multiclass classification with three classes (antonym, other, unrelated). Due to the skewed nature of the dataset, we combined category, entailment and synonym/paraphrases into one class. For both classification experiments, we perform random split with 70% train, 25% test, and 5% validation sets. Table 5 displays the number of relations in our dataset. Wikipedia[3] was used as the underlying corpus for all methods and we perform model selection on the validation set to tune the hyper-parameters of each method. We apply grid search for a range of values and pick the ones that yield the highest F_1 score on the validation set. The best hyper-parameters are reported in the appendix.

5.2 Baselines

Majority Baseline The majority baseline is achieved by labelling all the instances with the most frequent class occuring in the dataset i.e. FALSE (binary) or UNRELATED (multiclass).

[3]We used the English Wikipedia dump from May 2015 as the corpus.

Model	Binary			Multiclass		
	P	R	F_1	P	R	F_1
Majority baseline	0.304	0.551	0.392	0.222	0.472	0.303
SP baseline	0.661	0.568	0.436	0.583	0.488	0.344
Path-based SD baseline	0.723	0.724	0.722	0.636	0.675	0.651
Path-based AntNET	0.732	0.722	0.713	0.652	0.687	0.661**
Combined SD baseline	0.790	0.788	0.788	0.744	0.750	0.738
Combined AntNET	**0.803**	**0.802**	**0.802***	**0.746**	**0.757**	**0.746***

Table 6: Performance of the AntNET models in comparison to the baseline models.

Feature	Model	Binary			Multiclass		
		P	R	F_1	P	R	F_1
Distance	Path-based	0.727	0.727	0.724	0.665	0.692	0.664
	Combined	0.789	0.788	0.788	0.732	0.743	0.734
Negation	Path-based	0.732	0.722	0.713	0.652	0.687	0.661
	Combined	**0.803**	**0.802**	**0.802**	**0.746**	**0.757**	**0.746**

Table 7: Comparing the novel negation marking feature with the distance feature proposed by Nguyen et al. (2017).

Distributed Baseline The method proposed by Schwartz et al. (2015) uses symmetric patterns (SPs) for generating word embeddings. The authors automatically acquired symmetric patterns (defined as a sequence of 3–5 tokens consisting of exactly 2 wildcards and 1–3 words) from a large plain-text corpus, and generated vectors where each co-ordinate represented the co-occurrence in symmetric patterns of the represented word with another word of the vocabulary. For antonym representation, the authors relied on the patterns suggested by (Lin et al., 2003) to construct word embeddings containing an antonym parameter that can be turned on in order to represent antonyms as dissimilar, and that can be turned off to represent antonyms as similar. To evaluate the SP method on our data, we used the pre-trained SP embeddings[4] with 500 dimensions. We use the SVM classifier with RBF kernel for the classification of word pairs.

Path-based and Combined Baseline Since AntNET is an extension of the path-based and combined models proposed by (Shwartz and Dagan, 2016) for classifying multiple semantic relations, we use their models as additional baselines. Because their model used a different dataset that contained very few antonym instances, we replicated the baseline (SD) with the dataset and corpus information as in Sectionn 5.1 rather than comparing to the reported results.

5.3 Results

Table 6 displays the performance scores of AntNET and the baselines in terms of precision, recall and F_1. Our combined model significantly[5] outperforms all baselines in both binary and multiclass classifications. Both path-based and combined models of AntNET achieve a much better performance in comparison to the majority class and SP baselines.

Comparing the path-based methods, the AntNET model achieves a higher precision compared to the path-based SD baseline for binary classification, and outperforms the SD model in precision, recall and F_1 in the multiclass classification experiment. The low precision of the SD model stems from its inability to distinguish between antonyms and synonyms, and between related and unrelated pairs which are common in our dataset, causing many false positive pairs such as *difficult/harsh*, *bad/cunning*, *finish/far* which were classified as antonyms.

Comparing the combined models, the AntNET model outperforms the SD model in precision, recall and F_1, achieving state-of-the-art results for antonym detection. In all the experiments, the

[4] https://homes.cs.washington.edu/~roysch/papers/sp_embeddings/sp_embeddings.html

[5] We used paired t-test. *p $<$ 0.1, **p $<$ 0.05

performance of the model in the binary classification task was better than in the multiclass classification. Multiclass classification seems to be inherently harder for all methods, due to the large number of relations and the smaller number of instances for each relation. We also observed that as we increased the size of the training dataset used in our experiments, the results improved for both path-based and combined models, confirming the need for large-scale datasets that will benefit training neural models.

Effect of the Negation-marking Feature In our models, the novel negation marking feature is successfully integrated along the syntactic path to represent the paths between x and y. In order to evaluate the effect of our novel negation-marking feature for antonym detection, we compare this feature to the distance feature proposed by Nguyen et al. (2017). In their approach, they integrate the distance between related words in a lexico-syntactic path as a new pattern feature, along with lemma, POS and dependency for the task of distinguishing antonyms and synonyms. We re-implemented this model by making use of the same information regarding dataset and patterns as in Section 5.1 and then replacing the direction feature in the SD models by the distance feature.

The results are shown in Table 7 and indicate that the negation marking feature and the replacement of the embeddings of negated words by the ones of their base forms enhance the performance of our models more effectively than the distance feature does, across both binary and multiclass classifications. Although, the distance feature has previously been shown to perform well for the task of distinguishing antonyms from synonyms, this feature is not very effective in the multiclass setting.

5.4 Error Analysis

Figure 2 displays the confusion matrices for the binary and multiclass experiments of the best performing AntNET model. The confusion matrix shows that pairs were mostly assigned to the correct relation more than to any other class.

False Positives We analyzed the false positives from both the binary and multiclass experiments. We sampled about 20% false positive pairs and identified the following common errors. The majority of the misclassification errors stem from antonym-like or near-antonym relations: these are

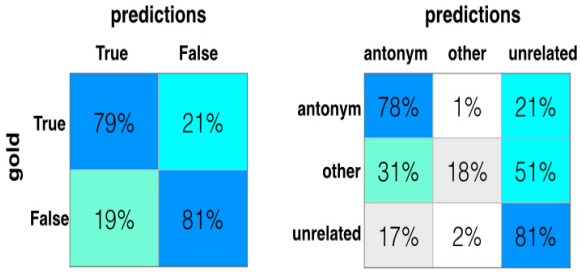

Figure 2: Confusion matrices for the combined AntNET model for binary (left) and multiclass (right) classifications. Rows indicate gold labels and columns indicate predictions. The matrix is normalized along rows, so that the predictions for each (true) class sum to 100%.

relations that could be considered as antonymy but were annotated by crowd-workers as other relations because they contain polysemous terms, for which the relation holds in a specific sense. For example: *north/south* and *polite/sassy* were labelled as *category* and *other* respectively. Other errors stem from confusing antonyms and unrelated pairs.

False Negatives We again sampled about 20% false positive pairs from both the binary and multiclass experiments and analyzed the major types of errors. Most of these pairs had only few co-occurrences in the corpus often due to infrequent terms (e.g. *cisc/risc* which define computer architectures). While our model effectively handled negative prefixes, it failed to handle negative suffixes causing incorrect classification of pairs like *spiritless/spirited*. A possible future work is to simply extend this model to handle negative suffixes as well.

6 Conclusion

In this paper, we presented an original technique for deriving antonyms using paraphrases from PPDB. We also proposed a novel morphology-aware neural network model, AntNET, which improves antonymy prediction for path-based and combined models. In addition to lexical and syntactic information, we suggested to include a novel morphological negation-marking feature.

Our models outperform the baselines in two relation classification tasks. We also demonstrated that the negation marking feature outperforms previously suggested path-based features for this task.

Since our proposed techniques for antonymy detection are corpus based, they can be applied to different languages and relations. The paraphrase-based method can be applied to other languages by extracting the paraphrases for these languages from the PPDB and using a morphological analysis tool (e.g. Morfette for French (Chrupala et al., 2008)) or by looking up the negation prefixes in a grammar book for languages that do not dispose of such a tool. The LSTM-based model could also be used in other languages since the method is corpus based, but we would need to create a training set for new languages. This would not however be too difficult; the training set used by the model is not that big (the one used here was around 6000 pairs) and could be easily labelled through crowdsourcing.

We release our code and the large-scale dataset derived from PPDB, annotated with semantic relations.

Acknowledgments

This material is based in part on research sponsored by DARPA under grant number FA8750-13-2-0017 (the DEFT program). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA and the U.S. Government.

This work has also been supported by the French National Research Agency under project ANR-16-CE33-0013 and partially supported by an Intel ICRI-CI grant, the Israel Science Foundation grant 880/12, and the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1).

We would like to thank our anonymous reviewers for their thoughtful and helpful comments.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*. Stroudsburg, PA, pages 597–604.

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL'01)*. Toulouse, France, pages 50–57.

Walter G. Charles and George A. Miller. 1989. Contexts of antonymous adjectives. *Applied Psychology* 10:357–375.

Grzegorz Chrupala, Georgiana Dinu, and Josef van Genabith. 2008. Learning Morphology with Morfette. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco, pages 2362–2367.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.

J. R. Firth. 1957. A synopsis of linguistic theory, 1930–1955. In *Studies in Linguistic Analysis*, Basil Blackwell, Oxford, United Kingdom, pages 1–32.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT)*. Atlanta, Georgia, pages 758–764.

Zellig S. Harris. 1954. Distributional structure. *Word* 10(23):146–162.

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*. Nantes, France, pages 539–545.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Constantine Lignos. 2010. Learning from Unseen Data. In *Proceedings of the Morpho Challenge 2010 Workshop*. Aalto University School of Science and Technology, Helsinki, Finland, pages 35–38.

Dekang Lin and Patrick Pantel. 2001. DIRT - Discovery of Inference Rules from Text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*. San Francisco, California, pages 323–328.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI '03)*. Acapulco, Mexico, pages 1492–1493.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)*. Lake Tahoe, Nevada, pages 3111–3119.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Distinguishing antonyms and synonyms in a pattern-based neural network. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL'17)*. Valencia, Spain, pages 76–85.

Ellie Pavlick, Johan Bos, Malvina Nissim, Charley Beller, Benjamin Van Durme, and Chris Callison-Burch. 2015a. Adding Semantics to Data-Driven Paraphrasing. In *The 53rd Annual Meeting of the Association for Computational Linguistics (ACL'15)*. Beijing, China, pages 1512–1522.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevich, and Chris Callison-Burch Ben Van Durme. 2015b. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL'15)*. Beijing, China, pages 425–430.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. Doha, Qatar, pages 1532–1543.

Michael Roth and Sabine Schulte im Walde. 2014. Combining Word Patterns and Discourse Markers for Paradigmatic Relation Classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL'14)*. Baltimore, MD, pages 524–530.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric Pattern Based Word Embeddings for Improved Word Similarity Prediction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning (CoNLL'15)*. Beijing, China, pages 258–267.

Vered Shwartz and Ido Dagan. 2016. CogALex-V Shared Task: LexNET - Integrated Path-based and Distributional Method for the Identification of Semantic Relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*. Osaka, Japan, pages 80–85.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*. Berlin, Germany, pages 2389–2398.

A Supplemental Material

For deriving antonyms using PPDB, we used the XXXL size of PPDB version 2.0 found in http://paraphrase.org/.

To compute the metrics in Tables 6 and 7, We used scikit-learn with the "averaged setup", which computes the metrics for each relation and reports their average weighted by support (the number of true instances for each relation). Note that it can result in a F_1 score that is not the harmonic mean of precision and recall.

During preprocessing we handled removal of punctuation. Since our dataset only contains short phrases, we removed any stop words occurring at the beginning of a sentence (Example: a man $\rightarrow$ man) and we also removed plurals. The best hyperparameters for all models mentioned in this paper are shown in Table 8. The learning rate was set to 0.001 for all experiments.

Model	Type	Dropout
SD-path	Binary	0.2
SD-path	Multiclass	0.4
SD-combined	Binary	0.4
SD-combined	Multiclass	0.2
ASD-path	Binary	0.0
ASD-path	Multiclass	0.2
ASD-combined	Binary	0.0
ASD-combined	Multiclass	0.2
AntNET-path	Binary	0.0
AntNET-path	Multiclass	0.2
AntNET-combined	Binary	0.4
AntNET-combined	Multiclass	0.2

Table 8: The best hyper-parameters in every model.