LEARNING ANTONYMS WITH PARAPHRASES AND A MORPHOLOGY-AWARE NEURAL NETWORK

Sneha Rajana

A THESIS

In

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial

Fulfillment of the Requirements for the Degree of Master of Science in Engineering

2017

_____

Prof. Chris Callison-Burch
Supervisor of Thesis

_____

Prof. Lyle Ungar
Thesis Reader

_____

Prof. Boon Thau-Loo
Masters Chairperson

Acknowledgements

I would like to thank Prof. Chris Callison-Burch for not only supervising this thesis but for being an amazing advisor and for getting me interested in NLP research.

I am thankful to Prof. Lyle Ungar for serving on the thesis committee and Prof. Boon Thau Loo and the CIS department of Penn Engineering for providing me with the encouragement and necessary resources to carry out this project.

Besides this, I would also like to thank Vered Shwartz, Bar-Ilan University, Israel for providing us with the preprocessed corpus and software for parts of this work and whose insightful advice and guidance led to the successful completion of this project.

I am grateful to my fellow Masters students whose constant encouragement and support was invaluable.

Finally, thank you to my family. I am who I am because of you.

Abstract

Recognizing and distinguishing antonyms from other types of semantic relations is a key part of language understanding systems and has widespread applications in Natural Language Processing tasks. In this study, we present two novel methods for identifying antonyms. In our first method, we use paraphrase pairs containing negation markers to derive antonym pairs. Using this technique, we created a dataset that is significantly larger than existing resources containing antonyms like WordNet and EVALution. In our second method, we propose a novel neural network model, *AntNET*, that integrates morphological features indicative of antonymy into a path-based relation detection algorithm. We demonstrate the effectiveness of these techniques with experimental results and show that AntNET outperforms state-of-the-art models for identifying and distinguishing antonyms from other semantic relations.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Semantics is a branch of linguistics which studies meaning in language or specifically "meaning relationships" between words. These meaning relationships can be defined by a number of different relations including, but not limited to synonymy, antonymy, hypernymy, hyponymy etc. Synonymy refers to words that are pronounced and spelled differently but contain the same meaning. For example, *happy* and *joyful* are synonyms of each other. Hypernymy and Hyponymy refers to a relationship between a general term and the more specific terms that fall under the category of the general term. For example, the birds *pigeon*, *sparrow*, and *crow* are hyponyms. They fall under the general term of *bird*, which is the hypernym.

Antonymy can be defined as the oppositeness of meaning between two expressions or expressions containing contrasting meanings. Over the years, linguists, cognitive scientists, psycholinguists, and lexicographers have tried to better understand and define antonymy. Palmer (1982) classified antonymy into the following three types.

- **Gradable antonymy** refers to a pair of words with opposite meanings where the two meanings lie on a continuous spectrum. The members of a pair differ in terms of *degree*. If something is not A, then it is not merely B, it can be any C or D or E in between A and B. For instance, the expression "today is not *hot*" may mean "today is not *cold*". There is a scale or a space exists between hot and cold, it may mean " today is *warm*". Other examples are *wet-dry*, *young-old*, *early-late*.

- **Complementary antonymy** refers to a pair of words with opposite meanings, where the two meanings do not lie on a continuous spectrum but have binary and contradictory meanings. If something is A, then it is not B. If something is X, then it is Y. The meaning of the word is absolute and not relative, there is only one possibility of meaning which is fixed, there is no intermediate ground between the members of a pair. If one is *dead*, one cannot be *alive*. Other examples are *on-off*, *alive-dead*, *entrance-exit*.

- **Relational antonymy** refers to a pair of words with opposite meanings, where opposite makes sense only in the context of the relationship between the two meanings. This is a special type of antonymy in which the members of a pair do not constitute a positive-negative opposition. They show the reversal of a relationship between two entities. X *buys* something from Y means the same as Y *sells* something to X. It is the same relationship seen from two different angles. Other examples are *parent-child*, *doctor-patient*, *give-receive*

In its strictest sense, antonymy applies to gradable adjectives, such as *hot-cold* and *tall-short*, where the two words represent the two ends of a semantic dimension. In a broader sense, it includes other adjectives, nouns, and verbs as well like *life-death*, *ascend-descend*, *shout-whisper*. In its broadest sense, it applies to any two words that represent contrasting meanings. The task of identifying antonymous expressions is valuable for NLP systems which go beyond recognizing semantic relatedness and require to identify specific semantic relations like synonymy, hypernymy etc. 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) proposed that antonyms occur together in a sentence

more often than chance. This is known as the co-occurrence hypothesis. 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 them 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 that occur 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; Shwartz and Dagan, 2016). They showed that a good path representation 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 to use 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 models 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.

1.1 Contributions

The main contributions of this thesis are:

- We present a novel technique of using paraphrases for antonym detection and successfully derive antonym pairs from paraphrases in the Paraphrase Database (Ganitkevitch et al., 2013; Pavlick et al., 2015b) (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.

1.2 Document Structure

The rest of this thesis is structures as follows. Chapter 2 comprises of literature review and goes over the related work in antonymy detection as well as work in identifying other semantic relations. In Chapetr 3, we describe our novel technique of deriving antonym pairs from paraphrases in PPDB and analyse and evaluate the derived pairs. In Chapter 4, we discuss AntNET, our morphology aware LSTM-based neural network model for identifying and separating antonyms from other semantic relations. Chapter 5 describes the different types of classification experiments, details of corpora and dataset used, and baseline models. Chapter 6 presents experimental results. It begins with preliminary experiments that motivate increasing the size of training corpora and preprocessing the training dataset. It describes the transformation of LexNET into AntNET with incremental models and new features and analyzes the effect of the negation marking features, the dataset, and word embeddings. Chapter 7 evaluates the performance of AntNET with baseline models and performs detailed error analysis. Chapter 8 discusses the implication of this work, and suggests future research directions.

Chapter 2

Literature Review

2.1 Degree of Antonymy

Mohammad and Hirst (2008) explored the relationship between what humans consider antonymous and how antonymy manifests itself in utterances. The study talks about 3 degrees of antonymy: strongly antonymous, semantically contrasting and not antonymous. The higher the degree of antonymy between target word pair, the higher the tendency to be considered antonym pairs by native human speakers.

Automatically determining the degree of antonymy between words can be helpful in detecting and generating paraphrases, detecting contradictions, detecting humor (satire and jokes tend to have contradictions and oxymorons) and in finding words which are semantically contrasting to a target word (probably to filter them out).

Antonymy, Synonymy, Hyponymy etc. are some lexical-semantic relations that apply to two lexical units - A combination of surface form and word sense. The study also explores the paradoxes of antonymy. Why are some pairs better antonyms? (Eg. large-small vs. large-little). Are semantic closeness and antonymy opposites? If two words are associated via synonymy, hyponymy-hypernymy or troponymy relations, they are considered to be semantically close or semantically related. Words that are semantically similar are also semantically related (Eg. plane-glider, doctor-surgeon) but not the other way round (Eg. plane-sky, surgeon-scalpel). Antonymous concepts are semantically re-

lated but not semantically similar. The co-occurrence hypothesis states that antonyms occur together in a sentence more often than chance (Charles and Miller 1989). But this is also true for hypernyms, holonyms, meronyms and near-synonyms. Thus, separating antonyms from them has proven to be difficult. Strong co-occurrence is not a sufficient condition for detecting antonyms, but it is useful. The distributional hypothesis of closeness states that words that occur in similar contexts tend to be semantically close.

Their study states that manually-created lexicons have limited coverage and do not include most semantically contrasting word pairs. They presented a new automatic and empirical measure of antonymy that combines corpus statistics with the structure of a published thesaurus. Their approach was as follows. The adjacency heuristic is that adjacent categories in most published thesauri are considered to be contrasting categories. Given a target word pair, the algorithm determined whether they are antonymous or not, and if they are, whether they have a high, medium, or low degree of antonymy. If the target words belong to the same thesaurus paragraphs as any of the seed antonyms linking the two contrasting categories, then the words have a high degree of antonymy. If the target words do not belong to the same thesaurus paragraphs as a seed antonym pair, but occur in contrasting categories, they have a low degree of antonymy if the word-pairs have a lower tendency to co-occur and a medium degree of antonymy if the word-pairs have a higher tendency to co-occur. This algorithm when evaluated on a set of closest-opposite questions, obtained a precision of over 80%.

2.2 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 by pivoting through foreign translations. This multilingual paraphrasing method often learns hypernym/hyponym pairs, e.g. 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) add interpretable semantics to PPDB and show that paraphrases in this resource represent a variety of entailment relations other than equivalence, including contradictory pairs like *nobody/someone* and *close/open*.

2.3 Semantic Taxonomy Induction

Snow et al. (2006) proposed a novel algorithm for inducing semantic taxonomies. Previous algorithms for taxonomy induction focused on independent classifiers for discovering single relationships based on hand-constructed or automatically generated textual patterns whereas their algorithm incorporates evidence from multiple classifiers over heterogeneous relationships to optimize the entire structure of the taxonomy. Though wide variety of relationship-specific classifiers like the pattern-based classifiers have achieved some degree of success, they frequently lack the global knowledge necessary to integrate their predictions into a complex taxonomy with multiple relations.

The paper mentions that previous algorithms focused only on inferring small taxonomies over a single relation, or has used evidence for multiple relations independently from one another. Another major shortfall was the inability to handle lexical ambiguity as these previous approaches sidestepped the issue of polysemy by making the assumption of only a single sense per word and inferring taxonomies explicitly over words and not senses. Their approach simultaneously provides a solution to the problems of jointly considering evidence about multiple relationships as well as lexical ambiguity within a single probabilistic framework. Within their model, they define the goal of taxonomy induction to be to find the taxonomy that maximizes the conditional probability of their

observations given the relationships of the taxonomy.

They have also extended their model to manage Lexical Ambiguity. If the objects in the taxonomy are word senses, they extended their model to allow for a many-to-many mapping (eg. word-to-sense mapping) between the the sets of objects. They have presented an algorithm for inducing semantic taxonomies which attempts to globally optimize the entire structure of the taxonomy. The models ability to integrate heterogeneous evidence from different classifiers offers a solution to the key problem of choosing the correct word sense to which to attach a new relation (hypernym, hyponym, antonym etc).

2.4 Natural Logic

The task of textual inference involves automatically determining whether a natural-language hypothesis can be inferred from a given premise. The NatLog system (MacCartney and Manning, 2007) which popularized natural logic for Rich Textual Entailment (RTE) tasks presented the first use of a computational model of natural logic - a system of logical inference operating over natural language for textual inference. Most current RTE systems achieve robustness by sacrificing semantic precision and those systems that rely on first-order logic and theorem proving are precise but excessively brittle. Their system found a low-cost edit sequence which transformed the premise into the hypothesis and learned to classify entailment relations across atomic edits. This system uses natural language as a representation and performs natural language inference using a structured algebra model. However, important kinds of inference like temporal reasoning, causal reasoning, paraphrasing and relation extraction are not addressed by natural logic.

2.5 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 bilingual dependency triples and 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 the word pairs (e.g. contrast relations might indicate antonymy and elaborations may indicate synonymy or hyponymy). Unlike previous pattern-based methods which used 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. In addition to the lexical and syntactic information, they also proposed the distance between the related words along the syntactic path as a new pattern feature.

2.6 RNNs for Relation Classification

Relation classification is a related task whose goal is to classify the relation that is expressed between two target terms in a given sentence to one of predefined relation classes. To illustrate, consider the following sentence, from the SemEval-2010 relation classification task dataset (Hendrickx et al., 2010): The $[apples]_{e1}$ are in the $[basket]_{e2}$. Here, the relation expressed between the target entities is Content Container(e1, e2). The shortest dependency paths between the target entities were shown to be informative for this task (Fundel et al., 2007). Recently, deep learning techniques showed good performance in capturing the indicative information in such paths. In particular, several papers show improved performance using recurrent neural networks (RNN) that process

a dependency path edge-by-edge. Xu et al. (2015) apply a separate long shortterm memory (LSTM) network to each sequence of words, POS tags, dependency labels and WordNet hypernyms along the path. A max-pooling layer on the LSTM outputs is used as the input of a network that predicts the classification. Other papers suggest incorporating additional network architectures to further improve performance (Nguyen and Grishman, 2015; Liu et al., 2015).

2.7 Integrated Pattern-based and Distributional Methods

In the past couple of years, deep learning models have been proposed for relation classification tasks. While Shwartz et al. (2016) first proposed their model to improve hypernymy detection between term-pairs, they later extended it to classify multiple semantic relations (Shwartz and Dagan, 2016), including antonyms. They suggested an improved path-based algorithm, in which the dependency paths are encoded using a recurrent neural network, that achieves results comparable to distributional methods. They then extended the approach to integrate both path-based and distributional signals in to the network, resulting in an improved performance for the semantic relation classification task. While their proposed model is very good at identifying relations like meronyms and hypernyms (state-of-the-art for hypernym detection), it does not perform too well in distinguishing between related and unrelated words, and between synonyms and antonyms. The morphology-aware neural network model that we propose handles these cases and better distinguishes antonyms from other semantic relations.

2.8 Vector Representation Methods

(Yih et al., 2012) introduced a new vector representation where antonyms lie on opposite sides of a sphere. they derived this representation with the incorporation of a thesaurus and latent semantic analysis, by assigning signs to the entries in the cooccur-

rence matrix on which latent semantic analysis operates, such that synonyms would tend to have positive cosine similarities, and antonyms would tend to have negative cosine similarities. Scheible et al. (2013) showed that the distributional difference between antonyms and synonyms can be identified via a simple word space model by using appropriate features. Instead of taking into account all words in a window of a certain size for feature extraction, the authors experimented with only words of a certain part-of-speech, and restricted distributions. Santus et al. (2014) proposed a different method to distinguish antonyms from synonyms by identifying the most salient dimensions of meaning in vector representations and reporting a new average-precision-based distributional measure and an entropy-based measure. Ono et al. (2015) trained supervised word embeddings for the task of identifying antonymy. They proposed two models to learn word embeddings: the first model relied on thesaurus information; the second model made use of distributional information and thesaurus information. More recently, Nguyen et al. (2016) proposed two methods to distinguish antonyms from synonyms: in the first method, the authors improved the quality of weighted feature vectors by strengthening those features that are most salient in the vectors, and by putting less emphasis on those that are of minor importance when distinguishing degrees of similarity between words. In the second method, the lexical contrast information was integrated into the skip-gram model (Mikolov et al., 2013) to learn word embeddings. This model successfully predicted degrees of similarity and identified antonyms and synonyms.

2.9 Modeling Multi-relational Data

Bordes et al. considered the problem of embedding entities and relationships of multi-relational data in low-dimensional vector spaces. They proposed a scalable, easy to train, canonical model with reduced parameters that models relationships by interpreting them as translations operating on the low-dimensional embeddings of the entities.

Multi-relational data refers to directed graphs whose nodes correspond to entities and edges of the form (head, label, tail) (denoted (h, l, t)), each of which indicates

that there exists a relationship of name label between the entities head and tail. Their work focused on modeling multi-relational data from Knowledge Bases (KBs), with the goal of providing an efficient tool to complete them by automatically adding new facts, without requiring extra knowledge. In contrast to single-relational data where ad-hoc but simple modeling assumptions can be made after some descriptive analysis of the data, the difficulty of relational data is that the notion of locality may involve relationships and entities of different types at the same time, so modeling multi-relational data requires more generic approaches that can choose the appropriate patterns considering all heterogeneous relationships at the same time.

In TransE, relationships are represented as translations in the embedding space: if (h, l, t) holds, then the embedding of the tail entity t should be close to the embedding of the head entity h plus some vector that depends on the relationship l. The main motivation behind their translation-based parameterization is that hierarchical relationships are extremely common in KBs and translations are the natural transformations for representing them. Since a null translation vector corresponds to an equivalence relationship between entities, this model can then represent the sibling relationship as well.

Their experiments demonstrate that this new model, despite its simplicity and its architecture primarily designed for modeling hierarchies, ends up being powerful on most kinds of relationships, and can significantly outperform state-of-the-art methods in link prediction on real world KBs.

Chapter 3

Paraphrase-based Antonym Derivation

In this chapter, we describe a novel automatic method of deriving antonym pairs from paraphrase pairs. Existing semantic resources like WordNET (Fellbaum, 1998) and EVALution (Enrico Santus and Huang, 2015) contain a much smaller set of antonyms compared to other semantic relations (e.g. synonyms, hypernyms and meronyms). Our aim is to create a large resource of high quality antonym pairs using paraphrases.

3.1 Resources

3.1.1 WordNet

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. Adjectives are organized in terms of antonymy. WordNet encodes antonymy as a lexical relationship a relation between two words and not concepts (Gross et al., 1989). WordNet antonym pairs comprise of "direct antonyms" (wet/dry, young/old) and "indirect antonyms" (dry/parched). Individual words across synsets are marked as direct antonyms. These pairs reflect the strong semantic contrast of their members. Each of these polar adjectives in turn is linked to a number of "semantically similar" ones. Semantically similar adjectives are indirect antonyms of the contral member of the op-

posite pole. Even after including the indirect antonyms, the coverage of WordNet is limited. WordNet or any other manually-created repository of antonyms does not encode the degree of antonymy between words (Mohammad and Hirst, 2008). Neverthess, we use WordNet to create a seed set of antonym pairs using a cross product of the synonyms and antonyms of words from WordNet and this was used as a gold standard in our experiments.

3.1.2 The Paraphrase Database (PPDB)

PPDB is an automatically extracted database containing millions of paraphrases in multiple languages. The goal of PPBD is to improve language processing by making systems more robust to language variability and unseen words. It is currently the largest available collection of paraphrases. PPDB contains over 150 million paraphrase rules covering three paraphrase types lexical (single word), phrasal (multiword), and syntactic restructuring rules. We focus on lexical and phrasal paraphrases up to two words in length, of which there are over half a million rules. The paraphrase database is released in six sizes (S, M, L, XL, XXL and XXXL) divided based on precision and recall scores. We have used the English PPDB, version 2.0 and XXXL size for all our experiments described later.

3.2 Antonym Derivation

3.2.1 Creation of a seed set of antonym pairs

We first created a seed set of antonyms generated using WordNet. As mentioned in Section 3.1.1, antonyms derived from WordNet are direct antonyms. We extended this list to include indirect antonym word pairs that were derived from a cross product of the synonyms of each word in the antonym pair and its direct antonym. Table 3.1 shows examples of direct antonym pairs and indirect antonym pairs. This seed set of antonym pairs generated from WordNet was used like a gold standard for further experiments.

Direct Antonyms	Indirect Antonyms
clean/dirty	clean/foul
rise/fall	rise/downfall
sleep/wake	sleep/rise
above/below	above/under

Table 3.1: Direct and Indirect antonyms retrieved from WordNet

3.2.2 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 these pairs to be the most indicative of an antonymy relationship between the target words. Table 3.2 shows examples of pairs in PPDB falling into these two categories.

Negating Word	Negating Prefix
not satisfactory/unsatisfactory	inadequate/ quite inadequate
not appropriate/inappropriate	unacceptable/ wholly unacceptable
not insignificant/significant	intolerable/ quite intolerable
not acceptable/objectionable	anti-discrimination/ non discrimination
not identical/different	deforestation/destruction

Table 3.2: Examples of paraphrase pairs from PPDB that were chosen for our experiments

3.2.3 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 MORSEL (Lignos, 2010) to perform morphological analysis and identify negation markers. For multi-word phrases beginning with a negating word, the negating word is simply dropped to obtain an antonym pair (e.g. different/*not* identical $\rightarrow$ different/identical). Our equation $ant(w, (p'_1, p_2))$ subsequently defines the antonym of a target word w and a paraphrase pair (p'_1, p_2)

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

belonging to the set of all selected paraphrase pairs P.

$$\forall_{(p_1',p_2)\in P \wedge w=p_1}[ant(w,(p_1',p_2))] = p_2 \tag{3.1}$$

p_1' is either a lexical phrase with a negating prefix or a multi-word phrasal pair beginning with not. The target word w is the base non-negated form of p_1' whose antonym is simply the paraphrase of p_1' or p_2. For a PPDB paraphrase pair $(unhappy/sad) \in P$, $antonym(happy) = sad$. Similarly, for the pair $(not \quad identical/different) \in P$, $antonym(identical) = different$.

Given a paraphrase pair $(p_1', p_2) \in P$, we derive the antonym pair (p_1, p_2) or (w, p_2) using Equation 3.1.

We also enrich the number of antonyms obtained using this technique by considering all synonyms (and lexical paraphrases) of p_2 ($u \in S(p_2)$) as antonyms of p_1 (or w) and synonyms of p_1 ($v \in S(p_1)$) as antonyms of p_2. Equation 3.2 describes this procedure. Given (p_1, p_2) derived from Equation 1:

$$\forall_{u\in S(p_2)}(ant(p_1)) = u \tag{3.2a}$$

$$\forall_{v\in S(p_1)}(ant(p_2)) = v \tag{3.2b}$$

In the above example, in addition to sad, we also retain its PPDB paraphrases and its WordNet synonyms as antonyms for $happy$.

In order to expand the antonym list, synonyms were obtained from WordNet and lexical paraphrases were obtained from PPDB. 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, unhappy/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). Some examples of PPDB paraphrase pairs and antonym pairs derived

from them are shown in Table 3.3. Table 3.4 shows the number of pairs derived at each step using PPDB. In total, we were able to derive around 213K unique pairs from PPDB. This is a much larger dataset than existing resources like WordNet and EVALution as shown in Table 3.5. Figure 3.1 displays the number of antonym pairs derived from each method explained above.

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

Table 3.3: Examples of antonyms derived from PPDB paraphrases. The antonym pairs in column 2 were derived from the corresponding paraphrase pairs in column 1.

Method	**#pairs**
(x,y) from paraphrase ($\tilde{x}$,y)/(x,$\tilde{y}$)	80,669
(x, paraphrase(y)), (paraphrase(x), y)	81,221
(x, synset(y)), (synset(x), y)	35,686

Table 3.4: Number of unique antonym pairs derived from PPDB at each step. Paraphrases and synsets were obtained from PPDB and WordNet respectively.

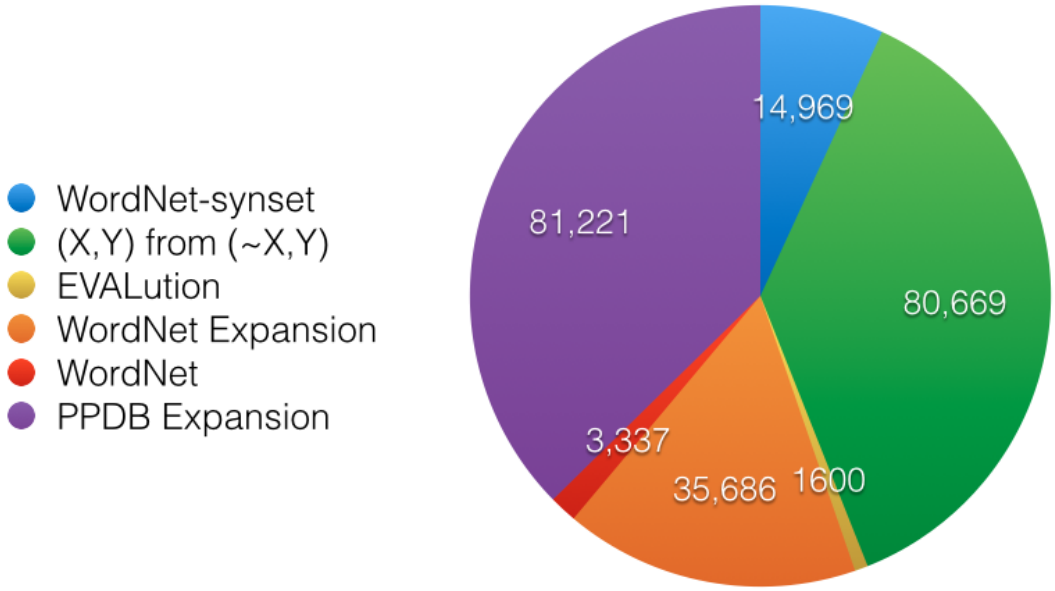

Figure 3.1: Illustration of the number of pairs derived as 'antonym' from different sources and methods.

Source	#pairs
EVALution	1,600
WordNet	18,306
PPDB	212,545

Table 3.5: 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.

3.3 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. A combined list of randomly selected antonyms from all of the methods listed above had about 53% accuracy. We also evaluated the percentage of antonyms yielded by each method. Figure 3.2 illustrates the percentage of antonyms derived from each method. Errors mostly consisted of phrases and words that do not have a 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 (habitants/general public). Table 3.6 gives some examples of categories of non-antonyms.

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

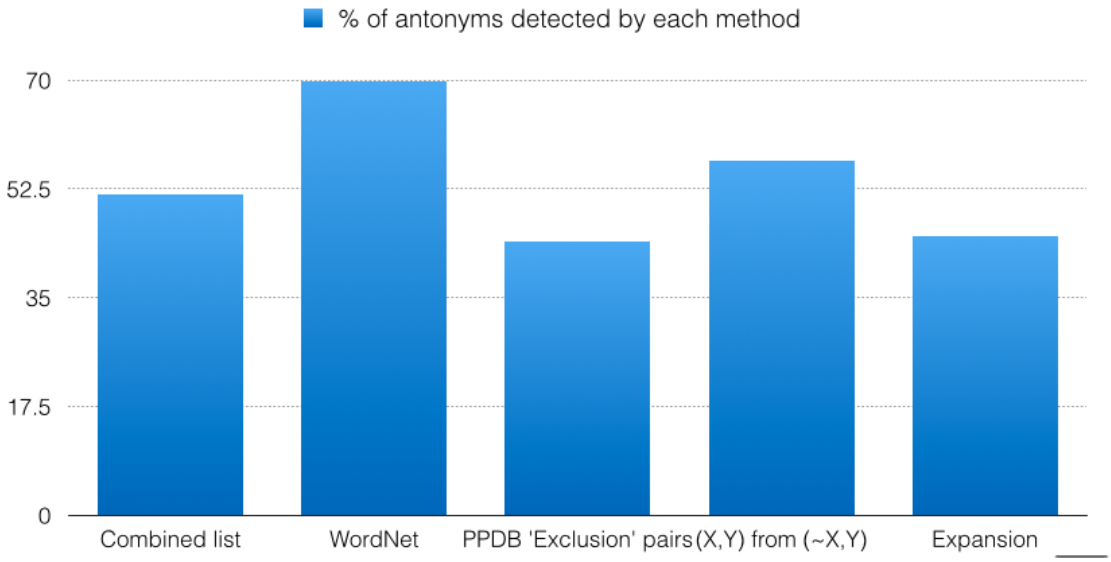

Figure 3.2: Illustration of the % of true antonyms from different sources and methods.

3.3.1 Hearst/Snow Patterns for Antonyms

In section 2.5 we described pattern based techniques to derive lexical relations from unrestricted text. Snow et al. (2006) and Hearst (1992) used easily recognizable and frequently occurring lexico-syntactic patterns to automatically derive lexical relations (hypernymy, hyponymy etc.) between noun pairs. Pavlick et al. (2015a) used monolingual path features to learn new patterns to differentiate between subtler relations like equivalence, negation and entailment. We used similar path features generated from dependency parses and Linguistic Data Consortium[2] (LDC) data to learn new patterns that are indicative of an antonym relationship. Table 3.7 gives examples of some of these paths.

Pattern	Example sentence
compared with X, Y	compared with the older generation, the new generation
to X or to Y	to fight or to surrender
X rather than Y	maximizing it rather than minimizing it
either X or Y	either low or high doses
neither X nor Y	neither women nor men

Table 3.7: Hearst/Snow paths for antonyms.

[2]https://www.ldc.upenn.edu/

3.4 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[3]. We asked workers to choose between the labels: antonym, synonym (or paraphrase for multi-word expressions), unrelated, other, entailment, and category. We show each pair to 7 workers, taking the majority label as truth.

[3] 5884 pairs were succesfully labelled by 13,434 annotators on www.crowdflower.com

Chapter 4

AntNET: a Morphology-aware Neural Network

In this chapter we describe AntNET, an LSTM-based, morphology aware neural network model for antonymy detection. We first focus on improving the neural embeddings of the path representation (Section 4.1), and then integrate distributional signals into this 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 (Hochreiter and Schmidhuber, 1997) (LSTM) network. 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 4.1. Figure 4.2 illustrates the differences in the path-

based architecture between LexNET and AntNET.

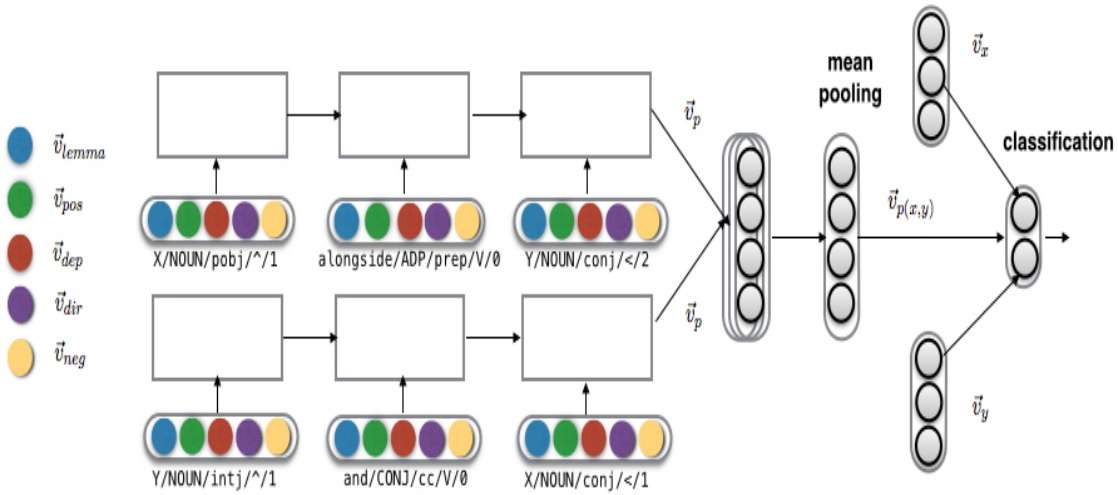

Figure 4.1: Illustration of the AntNET model. Each pair is represented by several paths and each path is a sequence of edges. An edge consists of five features: lemma, POS, dependency label, dependency direction, and negation marker.

4.1.1 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 my 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*

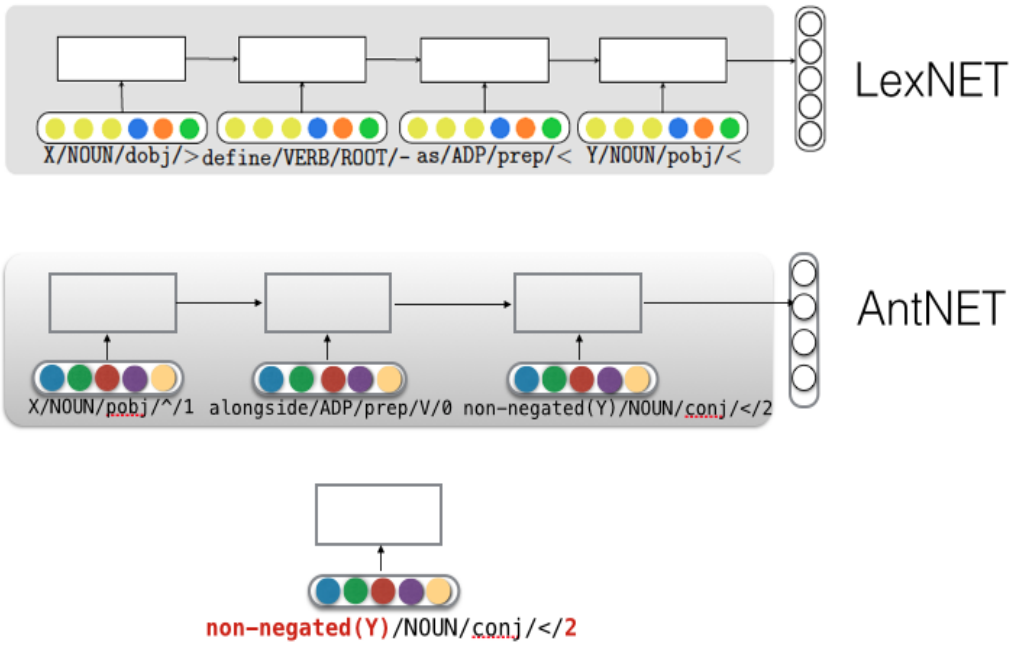

Figure 4.2: Comparing the path-based fetaures of LexNET and AntNET.

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.

4.1.2 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: $p = [\vec{edge}_1, \vec{edge}_2, ..., \vec{edge}_n,]$. The edge vectors are fed in order to an recurrent neural network (RNN) with LSTM units, resulting in the encoded path vector $\vec{v}_p$.

4.1.3 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:

$$\vec{v}_{p(x,y)} = \frac{\sum_{p \in P(x,y) f_p . \vec{v}_p}}{\sum_{p \in P(x,y) f_p}} \tag{4.1}$$

$\vec{v}_{p(x,y)}$ 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, 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:

$$c = softmax(MLP(\vec{v}_{p(x,y)}) \tag{4.2a}$$

$$r = argmax_i c[i] \tag{4.2b}$$

MLP stands for Multi Layer Perceptron, and can be computed with or without a hidden

layer (equations 4.3 and 4.4 respectively).

$$\vec{h} = tanh(W_1.\vec{v}_{p(x,y)} + b_1) \tag{4.3a}$$

$$MLP(\vec{v}_{p(x,y)}) = W_2.\vec{h} + b_2 \tag{4.3b}$$

$$MLP(\vec{v}_{p(x,y)}) = W_1.\vec{v}_{p(x,y)} + b_1 \tag{4.4}$$

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 chapter 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 $\vec{v}_{p(x,y)}$:

$$\vec{v}_{c(xy)} = [\vec{v}_x, \vec{v}_{p(x,y)}, \vec{v}_y] \tag{4.5}$$

Chapter 5

Experiments

For identifying antonymy, we experiment with the path-based and combined models of AntNET.

5.1 Types of Classification

5.1.1 Binary Classification

We first tried experimenting with binary classification with 2 labels True (for antonym pairs) and False (for non-antonym pairs). The dataset was split into 70% train, 25% test, and 5% validation sets. Hyper-parameters were tuned on the validation set to choose the best dropout rate, learning rate, GloVe embedding dimensions, and number of hidden layers.

5.1.2 Multiclass Classification

Six Classes

The first few experiments involved six labels Antonym, Category, Paraphrase, Unrelated, Entailment, and Other. The dataset was split into 70% train, 25% test, and 5% validation sets. Hyper-parameters were tuned on the validation set to choose the best dropout rate, learning rate, GloVe embedding dimensions, and number of hidden layers.

Three Classes

Given the skewed nature of the labels in the dataset, we thought combining some of the classes would help the model perform better. Category, Paraphrase, Entailment, and Other were clubbed into a single class Other. The final three classes were Antonym, Unrelated, and Other. The dataset was split into 70% train, 25% test, and 5% validation sets. Hyper-parameters were tuned on the validation set to choose the best dropout rate, learning rate, GloVe embedding dimensions, and number of hidden layers.

5.2 Resources

5.2.1 Dataset

Neural networks require a large amount of training data. we use the labelled portion of the dataset that we created using PPDB (Chapter 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, other, synonymy, category, unrelated) 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 and other-related pairs. Table 5.1 displays the number of relations in this dataset. Wikipedia[1] 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.

In order to show how our model performs in the notoriously difficult task of distinguishing antonyms and synonyms, we use the large-scale antonym and synonym

[1] we used the English Wikipedia dump from May 2015 as the corpus.

Train	Test	Val	Total
5122	1829	367	7318

Table 5.1: Number of instances present in the train/test/validation splits of the crowd-sourced dataset.

pairs from WordNet and Wordnik[2], previously used by Nguyen et al. (2016) for the same task. We use a 1:1 ratio of positive (antonym) to negative (synonym) pairs in the dataset. For both tasks, we perform random splitting with 70% train, 25% test, and 5% validation sets. Table 5.2 contains the number of pairs contained in the this datasets.

Word class	Train	Test	Val	Total
Verb	2534	908	182	3624
Noun	2836	1020	206	4062
Adjective	5562	1986	398	7946

Table 5.2: WordNET + Wordnik dataset

5.2.2 Corpus

The English Wikipedia dump from May 2015 was used as the corpus to train our integrated neural network model. The corpus is used to extract connecting dependency paths between target words. Paths were computed between the most frequent unigrams, bigrams, and trigrams in Wikipedia based on GloVe vocabulary and the most frequent 100K bigrams and trigrams. The vocabulary for the model consisted of PPDB words that were contained in the most common 400k words in Wikipedia and the most common 100k bigrams and trigrams in Wikipedia.

5.2.3 Word Embeddings

GloVe Embeddings GloVe stands for Global Vectors for Word Representation. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a

[2]http://www.wordnik.com

corpus, and the resulting representations showcase interesting linear substructures of the word vector space. We are using pre-trained word embeddings of different dimensions to train our model.

dLCE Embeddings Nguyen et al. (2016) proposed a novel extension of a skip-gram model with negative sampling (Mikolov et al., 2013) that integrates the lexical contrast information into the objective function of a skip-gram model. The proposed model optimizes the semantic vectors to predict degrees of word similarity and also to distinguish antonyms from synonyms. The improved word embeddings outperform state-of-the-art models on antonym synonym distinction and a word similarity task.

5.3 Baselines

5.3.1 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).

5.3.2 Distributed Baseline

The SP method proposed by Schwartz et al. (2015) uses symmetric patterns for generating word embeddings. the authors automatically acquire 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 generate 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 my data, we used the pre-trained SP embeddings[3] with 500 dimensions. we used the SVM classifier with

[3]http://homes.cs.washington.edu/ roysch/papers/ sp_embeddings/sp_embeddings.html

RBF kernel to perform for the classification of word-pairs.

5.3.3 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 chaptern 5.2.1 rather than comparing to the reported results.

5.3.4 Distributional Baseline

we apply the approach by Roth and Schulte im Walde (2014), henceforth RS. They used a vector space model to represent pairs of words by a combination of standard lexico-syntactic patterns and discourse markers. In addition to the patterns, the discourse markers added information to express discourse relations, which in turn may indicate the specific semantic relation between the two words in a word pair. For example, contrast relations might indicate antonymy, whereas elaborations may indicate synonymy or hyponymy.

Chapter 6

Results and Analysis

6.1 Preliminary Results

6.1.1 Effect of the dataset

Since LexNET was evaluated on a dataset that contained very few antonyms, our preliminary experiments included running LexNET on our dataset and making improvements to lead towards a better performance for the classification of antonyms. The first set of experiments were conducted on a small size of the dataset (722 pairs) that was manually labelled. In order to increase the size of the dataset, we crowdsourced the labelling task on CrowdFlower. This increased the size of the dataset to 5885 pairs but the dataset was skewed with an uneven distribution of classes. To fix this, we added antonyms from the EVALution dataset to the dataset generated using PPDB. The next set of experiments included preprocessing the data to handle punctuation within the words, analyzing false positives and false negatives and correcting the incorrectly labelled pairs in the dataset, and handling multi-word pairs. Table 6.1 shows the effect of the size of the dataset, crowdsourced labelling tasks, better label distribution and preprocessing data. All of these experiments saw a steady increase in the results for all types of classification.

Dataset size	Improvement	Binary			Multiclass(6)			Multiclass(3)		
		P	R	F_1	P	R	F_1	P	R	F_1
722	Self-labelling	0.78	0.77	0.77	0.64	0.70	0.66	0.72	0.71	0.69
5885	Crowdsourced-labelling	0.77	0.78	0.77	0.65	0.72	0.68	0.67	0.69	0.67
7453	Better label distribution	0.78	0.78	0.78	0.68	0.72	0.70	0.69	0.70	0.69
7318	Preprocessed data	0.80	0.80	0.80	0.71	0.76	0.73	0.75	0.74	0.72

Table 6.1: Effect of the dataset

6.1.2 Effect of the number of hidden layers

As explained in Section 4.1.3, the Multi Layer Perceptron for the final classification task can be computed with or without a hidden layer. In order to evaluate the effect of the number of hidden layers (0 for without and 1 for with a hidden layer), we repeat the above experiments by adding a hidden layer and compare the two sets of results. Table 6.2 shows the effect of the number of hidden layers.

Dataset size	Improvement	Binary			Multiclass(6)			Multiclass(3)		
		P	R	F_1	P	R	F_1	P	R	F_1
722	Self-labelling	0.80	0.79	0.79	0.62	0.65	0.63	0.67	0.67	0.66
5885	Crowdsourced-labelling	0.75	0.76	0.76	0.65	0.68	0.66	0.65	0.66	0.65
7453	Better label distribution	0.76	0.75	0.76	0.69	0.71	0.63	0.70	0.71	0.70
7318	Preprocessed data	0.78	0.78	0.78	0.72	0.74	0.72	0.70	0.71	0.70

Table 6.2: Repeating the previous experiments by adding a hidden layer

From the results of these experiments, we can see that in general, the models without a hidden layer performed better than those with a hidden layer across all experiments. It is possible that the contributions of the hidden layer and the path-based source over the distributional signal are redundant.

6.1.3 Effect of the number of dimensions of the word embeddings

In order to evaluate the effect of the dimensions of word embeddings and to choose the best one, we re-ran the first experiment with 722 pairs and self-labelled data (displayed in tables 6.1 and 6.2) with GloVe 100 and 200 dimensions. Table 6.3 shows the effect of the word embedding dimensions on our dataset.

From the results of these experiments, we can see that in general, the models with

GloVe dimension	# hidden layers	Binary			Multiclass(6)		
		P	**R**	**F_1**	**P**	**R**	**F_1**
50	0	0.78	0.77	0.77	0.64	0.70	0.66
	1	0.80	0.79	0.79	0.62	0.65	0.63
100	0	0.77	0.76	0.76	0.64	0.69	0.66
	1	0.75	0.75	0.75	0.62	0.64	0.63
200	0	0.76	0.76	0.75	0.62	0.65	0.63
	1	0.73	0.73	0.73	0.60	0.62	0.61

Table 6.3: Effect of GloVe dimensions

word embeddings of 50 dimensions performed the best, followed by 100 dimensions, and lastly, 200 dimensions.

6.2 Effect of the Negation-marking Feature

Based on these preliminary results, for further experiments leading to the development of our final models (AntNET-path and AntNET-combined), we use the dataset containing 7318 pairs for training, an MLP with no hidden layer, and GloVe embeddings of 50 dimensions. We also reduce the types of classification to binary and multiclass with 3 classes.

In our final model (AntNET), 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 variations of the AntNET model with slightly different features.

6.2.1 LexNET

Since AntNET is an extension of LexNET, we compare our novel negation feature with the path features of LexNET which include the POS tag, lemma, dependency label, and edge direction.

6.2.2 Negation Feature

In order to allow LexNET to better identify antonyms and better distinguish them from other semantic relations, we implemented AntNET-neg that adds a new morphological path-based feature to the existing features in LexNET. This new negation feature is used for marking whether the term pairs are negated.

6.2.3 Replacement of Word Embeddings

AntNET-morph is an improvement to AntNET-neg. AntNET-morph (or AntNET) not only records if either of the terms in the pair is negated but additionally, it also replaces the word (lemma) embeddings in the path by the word embedding of its base non-negated form.

6.2.4 Distance Feature

Nguyen et al. (2016) has previously shown that replacing the directino feature in HypeNET by the distance feature improves performance for the task of distinguishing between antonyms and synonyms. 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 labels. We re-implemented this model named AntNET-distance by making use of the same information regarding dataset and patterns as in chapter 5.2.1 and then replacing the direction feature in LexNET by the distance feature.

Model	**Binary**			**Multiclass**		
	P	**R**	**F_1**	**P**	**R**	**F_1**
LexNET (Path)	0.723	0.724	0.722	0.636	0.675	0.651
LexNET (Combined)	0.790	0.788	0.788	0.744	0.750	0.738
AntNET-distance (Path)	0.727	0.727	0.724	0.665	0.692	0.664
AntNET-distance-Combined	0.789	0.788	0.788	0.732	0.743	0.734
AntNET-neg (Combined)	0.798	0.798	0.798	0.738	0.750	0.740
AntNET-morph (Path)	0.732	0.722	0.713	0.652	0.687	0.661**
AntNET-morph (Combined)	**0.803**	**0.802**	**0.802***	**0.746**	**0.757**	**0.746***

Table 6.4: Effect of the novel negation-marking feature

The results are shown in Table 6.4 and indicate that the negation marking feature and the method of replacing the embeddings of negated words by their base forms, enhances the performance of our proposed models more effectively than the distance feature does, across both binary and multiclass classifications[1]. 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. Figure 6.1 compares the performance of the negation-marking feature with other features described above.

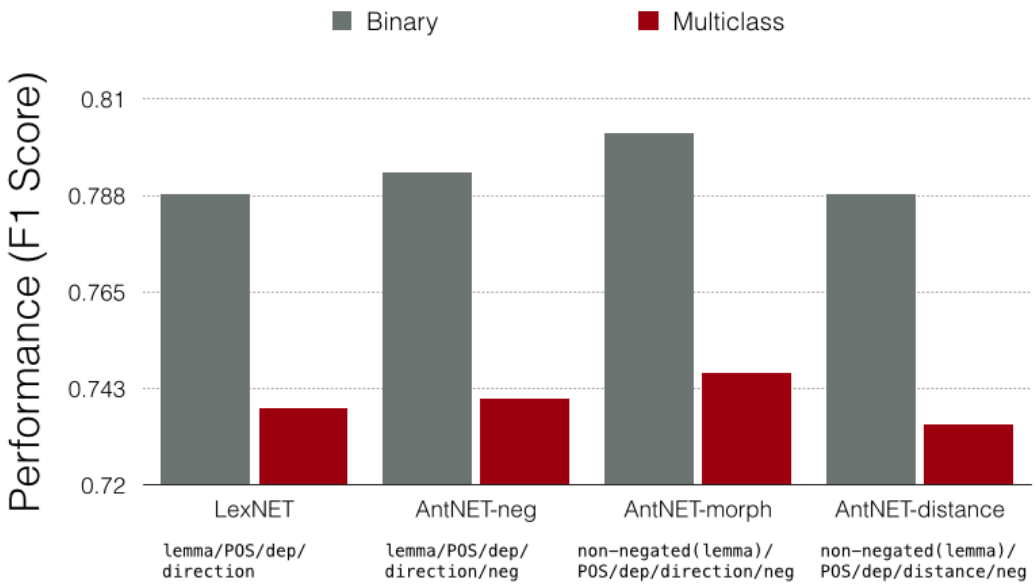

Figure 6.1: Illustration of the effect of the novel negation-marking feature.

6.3 Effect of Word Embeddings

Our methods rely on the GloVe word embeddings, state-of-the-art word embeddings for relation detection. In order to evaluate the effect of these word embeddings on the performance of our models, we replace them by the pre-trained dLCE embeddings with 100 dimensions, and compare the effects of the GloVe word embeddings and the dLCE word embeddings on the performance of AntNET. Nguyen et al. (2016) showed that the

[1] paired t-test, *p $<$ 0.1, **p $<$ 0.05

dLCE embeddings outperform state-of-the-art word embeddings for antonym-synonym distinction. Table 6.5 illustrates the performance of AntNET on binary classification experiments. The table shows that the pre-trained GloVe word embeddings are better than the pre-trained dLCE word embeddings, by around .02 F_1[2].

Word Embeddings	**P**	**R**	**F_1**
dLCE	0.784	0.783	0.784
GloVe	**0.803**	**0.802**	**0.802****

Table 6.5: Comparing pre-trained dLCE and GloVe word embeddings

[2]paired t-test, *p $<$ 0.1, **p $<$ 0.05

Chapter 7

Evaluation

In this chapter we evaluate the performance of AntNET with other relation detection models. Table 7.1 displays the performance scores of AntNET and the baselines, in terms of precision, recall, and F_1. my combined model significantly[1] outperforms all of the 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.

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 7.1: Performance of the AntNET models in comparison to the baseline models

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

[1] paired t-test, *p $<$ 0.1, **p $<$ 0.05

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

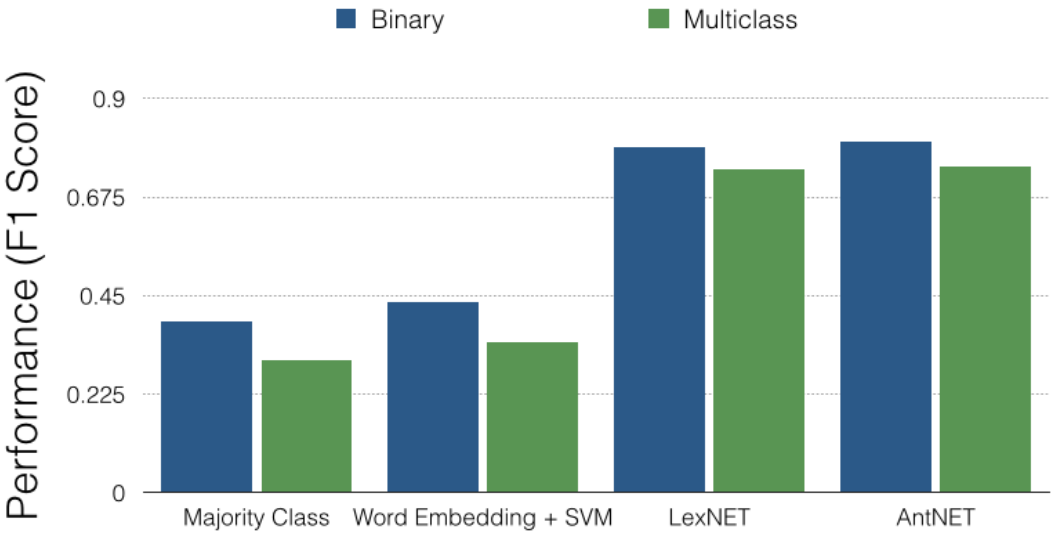

Figure 7.1: Illustration of the performance of AntNET with baselines.

7.1 Error Analysis

Figure 7.2 displays the confusion matrices for the binary and multiclass experiments and Figure 7.3 displays examples of pairs classified best performing AntNET model. The confusion matrix shows that pairs were mostly classified to the correct relation more than to any other class.

7.1.1 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 relations that could be considered as antonymy, but were annotated by crowd-workers as other relations because they contained 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.

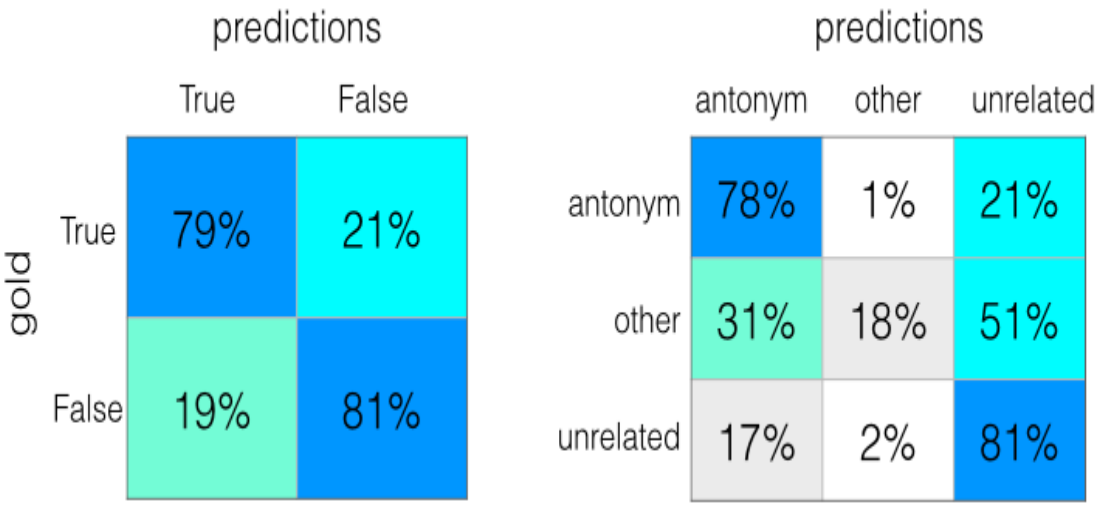

Figure 7.2: Confusion matrices for the combined AntNET model for binary (left) and multiclass (right) classifications. Rows indicate gold labels and columns indicate predictions. The matrix is normalized along rows, so that the predictions for each (true) class sum to 100%

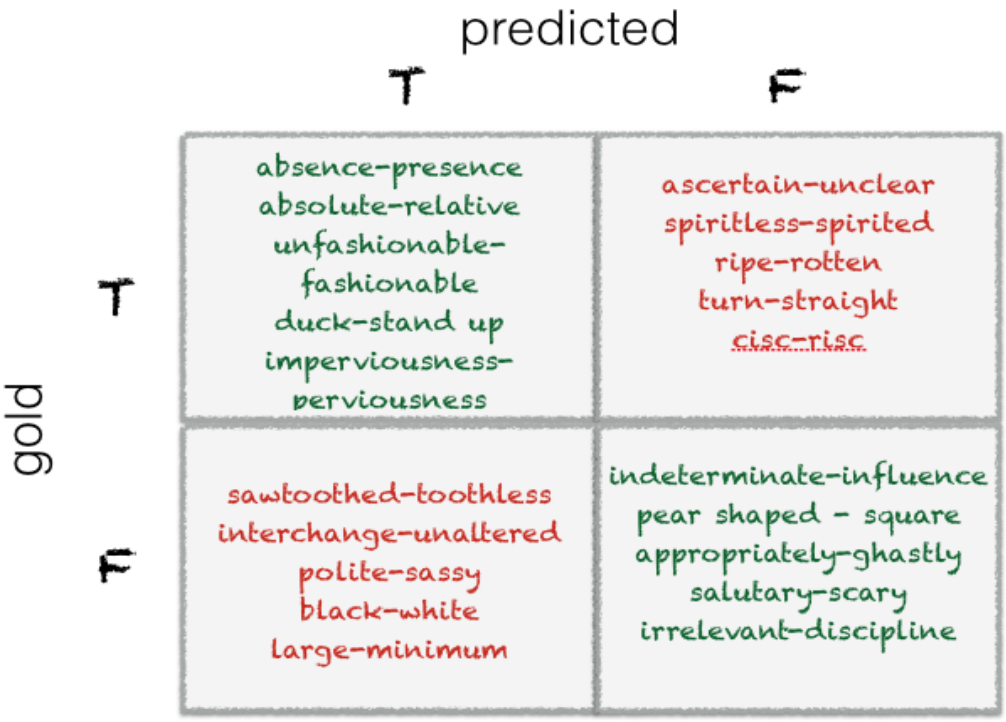

Figure 7.3: Example pairs classified by AntNET.

7.1.2 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 my 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.

7.1.3 Antonym-Synonym Distinction

Table 7.2 shows the performance scores of AntNET and the baseline methods according to the word classes (adjective, verb, and noun), in terms of precision (P), recall (R), and F_1.

Our model significantly outperforms the two baselines for adjectives and verbs,

Model	Adjective			Verb			Noun		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
SP baseline	0.730	0.706	0.718	0.560	0.609	0.584	0.625	0.393	0.482
RS baseline	0.717	0.717	0.717	0.789	0.787	0.788	**0.833**	**0.831**	**0.832**
AntNET-path	**0.752**	0.744	0.741	0.782	0.781	0.781	0.773	0.767	0.765
AntNET-comb	**0.752**	**0.752**	**0.752**	**0.799**	**0.793**	**0.792**	0.813	0.812	0.813

Table 7.2: Performance of the AntNET models compared to the baseline models for antonym-synonym distinction

achieving an improvement of 0.34 and 0.04 respectively, for $F1$. Regarding nouns, we do not outperform the more advanced RS baseline but in comparison to the SP baseline, my model still shows a clear F_1 improvement of 0.33. Distinguishing between antonyms and synonyms is challenging because they often occur in similar contexts. But with the help of the negation-marking feature, we were able to effectively distinguish between these pairs. wet is also possible that antonymous word pairs co-occur within a sentence more often than synonymous word pairs.

Chapter 8

Conclusion and Future Work

In this thesis, we presented an original technique for deriving antonyms using paraphrases from PPDB. we also presented 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 a novel morphological negation-marking feature.

Our proposed 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 my proposed techniques for antonymy detection are corpus based, they can be applied to different languages and relations.

For future work, we plan to annotate the rest of the dataset derived from PPDB by crowdsourcing the labelling task. We also plan to filter out the derived pairs to keep only those pairs where both the members of a pair belong to the same part-of-speech tag. Another filtering technique could be to test different PPDB 2.0 scores in order to choose the best threshold and keep only those pairs that have a higher score than the chosen threshold in PPDB.

Appendix A

Supplemental Material

Model	Type	Word 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 A.1: The best hyper-parameters in every model

To compute the metrics for evaluation, 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 an F_1 score that is not the harmonic mean of precision and recall.

While preprocessing we handled removal of punctuation. Since our dataset also contains short phrases, we removed any stop words occurring at the beginning of a sentence (Example: a man $\rightarrow$ man), and removing plurals. The best hyperparameters for all models mentioned in this paper are shown in Table A.1. The learning rate was set to 0.001 for all experiments

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '05, pages 597–604. https://doi.org/10.3115/1219840.1219914.

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '01, pages 50–57. https://doi.org/10.3115/1073012.1073020.

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

Frances Yung Alessandro Lenci Enrico Santus and Chu-Ren Huang. 2015. Evalution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics (LDL-2015)*. pages 64–69.

Christine Fellbaum. 1998. *WordNet*. Wiley Online Library.

John R. Firth. 1957. Long short-term memory. *Papers in Linguistics* pages 1934–51.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. Relex—relation extraction using dependency parse trees. *Bioinformatics* 23(3):365–371. https://doi.org/10.1093/bioinformatics/btl616.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*. Association for Computational Linguistics, Atlanta, Georgia, pages 758–764. http://cs.jhu.edu/ ccb/publications/ppdb.pdf.

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)*. Nantes, France, pages 539–545.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó. Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Stroudsburg, PA, USA, SemEval '10, pages 33–38. http://dl.acm.org/citation.cfm?id=1859664.1859670.

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

Constantine Lignos. 2010. Learning from unseen data. In Mikko Kurimo, Sami Virpioja, and Ville T. Turunen, editors, *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*. ACM, New York, NY, USA, KDD '01, pages 323–328. https://doi.org/10.1145/502512.502559.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*. pages 1492–1493. http://ijcai.org/Proceedings/03/Papers/249.pdf.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. *CoRR* abs/1507.04646.

Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Association for Computational Linguistics, Stroudsburg, PA, USA, RTE '07, pages 193–200. http://dl.acm.org/citation.cfm?id=1654536.1654575.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *CoRR* abs/1310.4546. http://arxiv.org/abs/1310.4546.

Saif Bonnie J. Dorr Mohammad and Graeme Hirst. 2008. Computing word-pair antonymy. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Waikiki, Honolulu, Hawaii, pages 982–991.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Thien Huu Nguyen and Ralph Grishman. 2015. Combining neural networks and log-linear models to improve relation extraction. *CoRR* abs/1511.05926.

Masataka Ono, Makoto Miwa, and Yutaka Sasaki. 2015. Word embedding-based antonym detection using thesauri and distributional information. In Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar, editors, *HLT-NAACL*. The Association for Computational Linguistics, pages 984–989. http://dblp.uni-trier.de/db/conf/naacl/naacl2015.htmlOnoMS15.

F.R. Palmer. 1982. *Semantics: A New Outline*. Cambridge University Press. https://books.google.com/books?id=FLBwQwAACAAJ.

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 2015). Beijing, China. http://www.cis.upenn.edu/ ccb/publications/adding-semantics-to-data-driven-paraphrasing.pdf.

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 2015)*. Association for Computational Linguistics, Beijing, China.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

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*. Baltimore, MD, pages 524–530.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte Im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*. pages 38–42.

Silke Scheible, Sabine Schulte Im Walde, and Sylvia Springorum. 2013. Uncovering distributional differences between synonyms and antonyms in a word space model. In *In Proceedings of the International Joint Conference on Natural Language Processing*. pages 489–497.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proceedings of CoNLL 2015*.

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), in COLING*. Osaka, Japan.

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 (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2389–2398. http://www.aclweb.org/anthology/P16-1226.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL-44, pages 801–808. https://doi.org/10.3115/1220175.1220276.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In Llus Mrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *EMNLP*. The Association for Computational Linguistics, pages 1785–1794. http://dblp.uni-trier.de/db/conf/emnlp/emnlp2015.htmlXuMLCPJ15.

Wen-tau Yih, Geoffrey Zweig, and John C. Platt. 2012. Polarity inducing latent semantic analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP-CoNLL '12, pages 1212–1222. http://dl.acm.org/citation.cfm?id=2390948.2391085.