# Learning to Distinguish Hypernyms and Co-Hyponyms

**Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir and Bill Keller**
Department of Informatics,
University of Sussex,
Brighton, UK
juliewe,D.Clarke,J.P.Reffin,davidw,billk@sussex.ac.uk

## Abstract

This work is concerned with distinguishing different semantic relations which exist between distributionally similar words. We compare a novel approach based on training a linear Support Vector Machine on pairs of feature vectors with state-of-the-art methods based on distributional similarity. We show that the new supervised approach does better even when there is minimal information about the target words in the training data, giving a 15% reduction in error rate over unsupervised approaches.

## 1 Introduction

Over recent years there has been much interest in the field of distributional semantics, drawing on the distributional hypothesis: words that occur in similar contexts tend to have similar meanings (Harris, 1954). There is a large body of work on the use of different similarity measures (Lee, 1999; Weeds and Weir, 2003; Curran, 2004) and many researchers have built thesauri (i.e. lists of "nearest neighbours") automatically and applied them in a variety of applications, generally with a good deal of success.

In early research there was much interest in how these automatically generated thesauri compare with human-constructed gold standards such as WordNet and Roget (Lin, 1998; Kilgarriff and Yallop, 2000). More recently, the focus has tended to shift to building thesauri to alleviate the sparse-data problem. Distributional thesauri have been used in a wide variety of areas including sentiment classification (Bollegala et al., 2011), WSD (Miller et al., 2012; Khapra et al., 2010), textual entailment (Berant et al., 2010), predicting semantic compositionality (Bergsma et al., 2010), acquisition of semantic lexicons (McIntosh, 2010), conversation entailment (Zhang and Chai, 2010), lexical substitution (Szarvas et al., 2013), taxonomy induction (Fountain and Lapata, 2012), and parser lexicalisation (Rei and Briscoe, 2013).

A primary focus of distributional semantics has been on identifying words which are similar to each other. However, semantic similarity encompasses a variety of different lexico-semantic and topical relations. Even if we just consider nouns, an automatically generated thesaurus will tend to return a mix of synonyms, antonyms, hyponyms, hypernyms, co-hyponyms, meronyms and other topically related words. A central problem here is that whilst most measures of distributional similarity are symmetric, some of the important semantic relations are not. The hyponymy relation (and converse hypernymy) which forms the ISA backbone of taxonomies and ontologies such as WordNet (Fellbaum, 1989), and determines lexical entailment (Geffet and Dagan, 2005), is asymmetric. On the other hand, the co-hyponymy relation which relates two words unrelated by hyponymy but sharing a (close) hypernym, is symmetric, as are synonymy and antonymy. Table 1 shows the distributionally nearest neighbours of the words `cat`, `animal` and `dog`. In the list for `cat` we can see 2 hypernyms and 13 co-hyponyms[1].

[1]We read `cat` in the sense `domestic cat` rather than `big cat`, hence `tiger` is a co-hyponym rather than hyponym of `cat`.





*Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland, August 23-29 2014.

| | |
|---|---|
| cat | dog 0.32, animal 0.29, rabbit 0.27, bird 0.26, bear 0.26, monkey 0.26, mouse 0.25, pig 0.25, snake 0.24, horse 0.24, rat 0.24, elephant 0.23, tiger 0.23, deer 0.23, creature 0.23 |
| animal | bird 0.36, fish 0.34, creature 0.33, dog 0.31, horse 0.30, insect 0.30, species 0.29, cat 0.29, human 0.28, mammal, 0.28, cattle 0.27, snake 0.27, pig 0.26, rabbit 0.26, elephant 0.25 |
| dog | cat 0.32, animal 0.31, horse 0.29, bird 0.26, rabbit 0.26, pig 0.25, bear 0.26, man 0.25, fish 0.24, boy 0.24, creature 0.24, monkey 0.24, snake 0.24, mouse 0.24, rat 0.23 |

Table 1: Top 15 neighbours of `cat`, `animal` and `dog` generated using Lin's similarity measure (Lin, 1998) considering all words and dependency features occurring 100 or more times in Wikipedia.

Distributional similarity is being deployed (e.g., Dinu and Thater (2012)) in situations where it can be useful to be able to distinguish between these different relationships. Consider the following two sentences.

*The cat ran across the road.* (1)

*The animal ran across the road.* (2)

Sentence 1 textually entails sentence 2, but sentence 2 does not textually entail sentence 1. The ability to determine whether entailment holds between the sentences, and in which direction, depends on the ability to identify hyponymy. Given a similarity score of 0.29 between `cat` and `animal`, how do we know which is the hyponym and which is the hypernym?

In applying distributional semantics to the problem of textual entailment, there is a need to generalise lexical entailment to phrases and sentences. Thus, the ability to distinguish different semantic relations is crucial if approaches to the composition of distributional representations of meaning that are currently receiving considerable interest (Widdows, 2008; Mitchell and Lapata, 2008; Baroni and Zamparelli, 2010; Grefenstette et al., 2011; Socher et al., 2012; Weeds et al., 2014) are to be applied to the textual entailment problem.

We formulate the challenge as follows: Consider a set of pairs of similar words $\langle A, B \rangle$ where one of three relationships hold between $A$ and $B$: $A$ lexically entails $B$, $B$ lexically entails $A$ or $A$ and $B$ are related by co-hyponymy. Given such a set, how can we determine which relationship holds? In Section 2, we discuss existing attempts to address this problem through the use of various *directional* measures of distributional similarity.

This paper considers the effectiveness of various supervised approaches, and makes the following contributions. First, we show that a SVM can distinguish the entailment and co-hyponymy relations, achieving a significant reduction in error rate in comparison to existing state-of-the-art methods based on the notion of distributional generality. Second, by comparing two different data sets, one built from BLESS (Baroni and Lenci, 2011) and the other from WordNet (Fellbaum, 1989), we derive important insights into the requirements of a valid evaluation of supervised approaches, and provide a data set for further research in this area. Third, we show that when learning how to determine an ontological relationship between a pair of similar words by means of the word's distributional vectors, quite different vector operations are useful when identifying different ontological relationships. In particular, using the difference between the vectors for pairs of words is appropriate for the entailment task, whereas adding the vectors works well for the co-hyponym task.

## 2 Related Work

Lee (1999) noted that the substitutability of one word for another was asymmetric and proposed the alpha-skew divergence measure, an asymmetric version of the Kullback-Leibler divergence measure. She found that this measure improved results in language modelling, when a word's distribution is smoothed using the distributions of its nearest neighbours.

Weeds et al. (2004) proposed a notion of distributional generality, observing that more general words tend to occur in a larger variety of contexts than more specific words. For example, we would expect to be able to replace any occurrence of `cat` with `animal` and so all of the contexts of `cat` must be plausible

contexts for `animal`. However, not all of the contexts of `animal` would be plausible for `cat`, e.g., "the monstrous animal barked at the intruder". Weeds et al. (2004) attempt to capture this asymmetry by framing word similarity in terms of co-occurrence retrieval (Weeds and Weir, 2003), where precision and recall are defined as:

$$P_{ww}(u,v) = \frac{\Sigma_{f \in F(u) \cap F(v)} I(u,f)}{\Sigma_{f \in F(u)} I(u,f)} \text{ and } R_{ww}(u,v) = \frac{\Sigma_{f \in F(u) \cap F(v)} I(v,f)}{\Sigma_{f \in F(v)} I(v,f)}$$

where $I(n,f)$ is the pointwise mutual information (PMI) between noun $n$ and feature $f$ and F(n) is the set of all features $f$ for which $I(n,f) > 0$.

By comparing the precision and recall of one word's retrieval of another word's contexts, they were able to successfully identify the direction of an entailment relation in 71% of pairs drawn from WordNet. However, this was not significantly better than a baseline which proposed that the most frequent word was the most general.

Clarke (2009) formalised the idea of distributional generality using a partially ordered vector space. He also argued for using a variation of co-occurrence retrieval where precision and recall are defined as:

$$P_{cl}(u,v) = \frac{\Sigma_{f \in F(u) \cap F(v)} min(I(u,f), I(v,f))}{\Sigma_{f \in F(u)} I(u,f)} \text{ and } R_{cl}(u,v) = \frac{\Sigma_{f \in F(u) \cap F(v)} min(I(u,f), I(v,f))}{\Sigma_{f \in F(v)} I(v,f)}$$

Lenci and Benotto (2012) took the notion further and hypothesised that more general terms should have high recall and low precision, which would thus make it possible to distinguish them from other related terms such as synonyms and co-hyponyms. They proposed a variant of the Clarke (2009) measure to identify hypernyms:

$$invCL(u,v) = \sqrt[2]{P_{cl}(u,v) * (1 - R_{cl}(u,v))}$$

Evaluation on the BLESS data set (Baroni and Lenci, 2011), showed that this measure is better at distinguishing hypernyms from other relations than the measures of Weeds et al. (2004) and Clarke (2009).

Geffet and Dagan (2005) proposed an approach based on *feature inclusion*, which extends the rationale of Weeds et al. (2004) to lexical entailment. Using data from the web they demonstrated a strong correlation between complete inclusion of prominent features and lexical entailment. However, they were unable to assess this using an off-line corpus due to data sparseness.

Szpektor and Dagan (2008) found that the $P_{ww}$ measure tends to promote relationships between infrequent words with narrow vectors (i.e. those with relatively few distinct context features). They proposed using the geometric average of $P_{ww}$ and the symmetric similarity measure of Lin (1998) in order to penalise low frequency words.

Kotlerman et al. (2010) apply the IR evaluation method of *Average Precision* to the problem of identifying lexical inference and use the balancing approach of Szpektor and Dagan (2008) to demote similarities for narrow feature vectors; their measure is called *balAPinc*. They show that all of the asymmetric similarity measures previously proposed perform much better than symmetric similarity measures on a directionality detection experiment, and that their method and that of Clarke (2009) outperform the others with statistical significance. They also show that their measure is superior when used for term expansion in an event detection task.

Baroni et al. (2012) investigate the relation between phrasal and lexical entailment, and demonstrate that support vector machines can generalise entailment relations between quantifier phrases to entailment involving unseen quantifiers. They compare the performance of their system with the *balAPinc* measure.

The Stanford WordNet project (Snow et al., 2004) expands the WordNet taxonomy by analysing large corpora to find patterns that are indicative of hyponymy. For example, the pattern "$NP_X$ and other $NP_Y$" is an indication that $NP_X$ is a $NP_Y$, i.e. that $NP_X$ is a hyponym of $NP_Y$. They use machine learning to identify other such patterns from known hyponym-hypernym pairs, and then use these patterns to find new relations in the corpus. The transitivity relation of the taxonomy is enforced by searching only over valid taxonomies and evaluating the likelihood of each taxonomy given the available evidence (Snow

et al., 2006). The approach is similar to ours in providing a supervised method of learning semantic relations, but relies on having features for occurrences of pairs of terms rather than just vectors for terms themselves. Our approach is therefore more generally applicable to systems which compose distributional representations of meaning.

Most recently, Rei and Briscoe (2013) note that hyponyms are well suited for lexical substitution. In their experiments with smoothing edge scores for parser lexicalisation, they find that a directional similarity measure, *WeightedCosine*[2], performs best. Also of note, Mikolov et al. (2013) propose a vector offset method to capture syntactic and semantic regularities between word representations learnt by a recurrent neural network language model. Yih et al. (2012) present a method for distinguishing synonyms and antonyms by inducing polarity in a document-term matrix before applying Latent Semantic Analysis. Santus et al. (2014) propose identifying hypernyms using a new measure based on entropy, SLQS, which is based on the hypothesis that the most typical linguistic contexts of a hypernym are less informative than the most typical linguistic contexts of its hyponyms. Evaluated on pairs extracted from the BLESS dataset (Baroni and Lenci, 2011), this measure outperforms $P_{ww}$ at both discriminating hypernym test pairs from other types of relation and at determining the direction of the entailment relation.

## 3 Methodology

The code used to perform our experiments has been open sourced, and is available online.[3]

### 3.1 Vector Representations

Distributional information was collected for all of the nouns from Wikipedia provided they had occurred 100 or more times. We used a Wikimedia dump of Wikipedia from June 2011 and extracted text using wp2txt[4]. This was part-of-speech tagged, lemmatised and dependency parsed using the Malt Parser (Nivre, 2004). All major grammatical dependency relations involving open class parts of speech (`nsubj, dobj, iobj, conj, amod, nnmod`) and also occurring 100 or more times were extracted as features of the POS-tagged and lemmatised nouns. The value of each feature is the positive point wise mutual information (PPMI) (Church and Hanks, 1989) between the noun and the feature. The total number of noun vectors which can be harvested from Wikipedia with these parameters is $124,345$.

Our goal is to build classifiers that establish whether or not a given semantic relation, *rel*, holds between two similar words $A$ and $B$. Support vector machines (SVMs), which are effective across a variety of classification scenarios, learn a boundary between two classes from a set of positive and negative example vectors. The two classes correspond to the relation *rel* holding or not holding. Here, however, we do not start with a single vector, but with two distributional vectors $v_A$ and $v_B$ for the words $A$ and $B$, respectively. These vectors must be combined in some way to produce the SVM's input, and a number of ways were considered, defined in Table 2. Of these operations, the vector difference (used by *svmDIFF* and *knnDIFF*) and direct sum (used by *svmCAT*) are asymmetric, whereas the sum and pointwise multiplication (used by *svmADD* and *svmMULT*) are symmetric.

We now motivate the use of each of these operations. First, we note that pointwise multiplication (*svmMULT*) is intersective. Similar vectors will have a large intersection and it might be possible to learn the features that nouns occurring in different semantic relations should share. However, it does not retain any information about non-shared features and it is symmetric so it is difficult to see how it would be possible to use it to distinguish hypernyms from hyponyms. Pointwise addition (*svmADD*) effectively performs the union of the features, giving emphasis to the shared features. Whilst it does retain information about the non-shared features, it is also symmetric, making it difficult again to see how it would be useful in determining the direction of an entailment relation

Vector difference (as used in *svmDIFF* and *knnDIFF*), on the other hand, is asymmetric. Further, we might expect a small difference vector (containing many zeroes) to be indicative of similar nouns. Further, considering the majority sign of features in this difference vector might indicate the direction of

[2]The details of this measure are unpublished.

[3]`https://github.com/SussexCompSem/learninghypernyms`

[4]`https://github.com/yohasebe/wp2txt`

entailment. Using an SVM, we might expect to be able to effectively learn which of these features should be ignored and which should be combined, to decide the correct direction of entailment in the majority number of cases in our training data. However, note that if one uses vector difference it is impossible to distinguish between the case where a feature occurred with both nouns (to the same extent) and the case where a feature occurs with neither noun. Accordingly, a small difference vector may indicate that both nouns do not occur in many distinct contexts. A possible solution to this problem is to use the direct sum of the vectors (i.e., the concatenation of the two vectors) which retains all of the information from the original vectors. Finally, we consider the use of the single vector corresponding to the second word (*svmSING*) as a baseline. High performance by this operation would indicate that we can learn features of words which tend to be hypernyms (or co-hyponyms) without any regard to the other word in the putative relationship.

We also note that the behaviour of these methods may differ depending on the weighting used for vectors. For example, PMI is the log of a ratio of probabilities and therefore one might expect vector addition where vectors are weighted using PMI to correspond to multiplication where vectors are weighted using frequency or probability. However, the use of *positive* PMI (where negative PMI scores are regarded equal to zero), which is consistent with other work in this area, means that this correspondence is lost.

Because of the nature of our datasets, we were concerned that systems could learn information about the taxonomy from the relations in the training data, without making use of information in the vectors themselves. To investigate this, we constructed random vectors to be used in place of the vectors derived from Wikipedia. The dimensionality of the random vectors was chosen to be 1000 since this substantially exceeds the average number (398) of non-zero features in the Wikipedia vectors.

### 3.2 Classifiers

We constructed linear SVMs for each of the vector operations outlined in Section 3.1. We used linear SVMs for speed and simplicity, since the point is to compare the different vector representations of the pairings. For comparison, we also constructed a number of supervised, unsupervised, and weakly supervised classifiers. These are listed in Table 2. For the linear SVMs and kNN classifier, we used the scikit-learn implementations with default settings. For $k$ nearest neighbours, we performed a parameter search, using nested cross-validation, varying $k$ between 1 and 50.

For weakly supervised approaches, we evaluated the measure on the training set, then found the best threshold $p$ on the training set that best divides the two classes using that measure. When classifying, we determine that the relation holds if the value of the measure exceeds $p$.

| | |
|---|---|
| *svmDIFF* | A linear SVM trained on the vector difference $v_B - v_A$ |
| *svmMULT* | A linear SVM trained on the pointwise product vector $v_B * v_A$ |
| *svmADD* | A linear SVM trained on the vector sum $v_B + v_A$ |
| *svmCAT* | A linear SVM trained on the vector concatenation $v_B \oplus v_A$ |
| *svmSING* | A linear SVM trained on the vector $v_B$ |
| *knnDIFF* | $k$ nearest neighbours (knn) trained on the vector difference $v_B - v_A . 1 < k < 50$ |
| *widthdiff* | $\mathit{width}(B) > \mathit{width}(A) \rightarrow \mathit{rel}(A, B)$ where $\mathit{width}(A)$ is number of non-zero features in $A$ |
| *singlewidth* | $\mathit{width}(B) > p \rightarrow \mathit{rel}(A, B)$ |
| *cosineP* | $\mathit{sim}_{cos}(A, B) > p \rightarrow \mathit{rel}(A, B)$ where $\mathit{sim}_{cos}(A, B)$ is cosine similarity using PPMI |
| *linP* | $\mathit{sim}_{lin}(A, B) > p \rightarrow \mathit{rel}(A, B)$ (Lin, 1998) |
| *CRdiff* | $P_{ww}(A, B) > R_{ww}(A, B) \rightarrow \mathit{rel}(A, B)$ (Weeds et al., 2004) |
| *clarkediff* | $P_{cl}(A, B) > R_{cl}(A, B) \rightarrow \mathit{rel}(A, B)$ (Clarke, 2009) |
| *invCLP* | $\mathit{invCL}(A, B) > p \rightarrow \mathit{rel}(A, B)$ (Lenci and Benotto, 2012) |
| *balAPincP* | $\mathit{balAPinc}(A, B) > p \rightarrow \mathit{rel}(A, B)$ (Kotlerman et al., 2010) |
| *most freq* | The most frequent label in the training data is assigned to every test point. |

Table 2: Implemented classifiers

### 3.3 Data Sets

One of key the challenges of this work has been to construct a data set which accurately and validly tests our hypotheses. All four of our datasets detailed below are available online [5].

In order to test our hypotheses, a data set needs to be balanced in many respects in order to prevent the supervised classifiers making use of artefacts of the data. This would not only make it unfair to compare the supervised approaches with the unsupervised approaches, but also make it unlikely that our results would be generalisable to other data. Here, we outline the requirements for the data sets, the importance of which is demonstrated by our initial results for a data set which does not satisfy all of them.

There should be an equal number of positive and negative examples of a semantic relation. Thus, random guessing or labelling with the most frequently seen label in the training data will yield 50% accuracy and precision. An advantage of incorporating this requirement means that evaluation can be in terms of simple accuracy (or error rate).

It should not be possible to do well simply by considering the distributional similarity of the terms. Hence, the negative examples need to be pairs of equally similar words, but where the relationship under consideration does not hold.

It should not be possible to do well by pre-supposing an entailment relation and guessing the direction. For example, it has been shown (Weeds et al., 2004) that given a pair of entailing words selected from WordNet, over 70% of the time the more frequent word is also the entailed word.

It should not be possible to do well using ontological information learnt about one or both of the words from the training data that is not generalisable to their distributional representations. For example, it should not be possible for the classifier simply to learn directly from the training pairs ⟨`cat` ISA `mammal`⟩ and ⟨`mammal` ISA `animal`⟩ that ⟨`cat` ISA `animal`⟩. Furthermore, we must ensure that a classifier cannot learn that a particular word is near the top of the ontological hierarchy, and, as a result, do well by guessing that a particular pairing probably has an entailment relation. For example, given many pairs such as ⟨`cat` ISA `animal`⟩, ⟨`dog` ISA `animal`⟩, a system which guessed ⟨`rabbit` ISA `animal`⟩ but not ⟨`animal` ISA `rabbit`⟩ would do better than random guessing. Whilst both of these types of information could be useful in a hybrid system, they do not require any distributional information and therefore we would not be learning anything about the distributional features of `animal` which make it likely to be a hypernym.

#### 3.3.1 BLESS

We have constructed two data sets from BLESS (Baroni and Lenci, 2011) which is a collection of examples of hypernyms, co-hyponyms, meronyms and random unrelated words for each of 200 concrete, largely monosemous nouns. We will refer to these 200 nouns as the BLESS concepts.

$hyponym_{\text{BLESS}}$ is a set of 1976 labelled pairs of nouns. For each BLESS concept, 80% of the hypernyms were randomly selected to provide positive examples of entailment. The remaining hypernyms for the given concept were reversed and taken with the same number of co-hyponyms, meronyms and random words to form negative examples of entailment. A filter was applied to ensure that duplicate pairs were not included (e.g., if ⟨`cat`, `animal`⟩ is a positive pair then ⟨`animal`, `cat`⟩ cannot be a negative pair).

$cohyponym_{\text{BLESS}}$ is a set of 5835 labelled pairs of nouns. For each BLESS concept, the co-hyponyms were taken as positive examples of this relation. The same total number of (and split evenly between) hypernyms, meronyms and random words was taken to form the negative examples. The order of 50% of the pairs was reversed and again duplicate pairs were disallowed.

In both cases the pairs are labelled as positive or negative for the specified semantic relation and in both cases there are equal ($\pm 1$) numbers of positive and negative examples. For 99% of the generated BLESS pairs, both nouns had associated vectors harvested from Wikipedia. If a noun does not have an associated vector, the classifiers use a zero vector.

[5] `https://github.com/SussexCompSem/learninghypernyms`

### 3.3.2 WordNet

We constructed two data sets using WordNet. Whilst these data sets are similar in size to the BLESS data sets they more adequately satisfy the requirements laid out above[6]. We constructed a list of all non-rare, largely monosemous, single word terms in WordNet. To be considered non-rare, a word needed to have occurred in SemCor at least once (i.e. frequency information is provided about it in the WordNet package) and to have occurred in Wikipedia at least 100 times. To be considered largely monosemous, the predominant sense of the word needed to account for over 50% of the occurrences in the SemCor frequency information provided with WordNet. This led to a list of 7613 nouns.

*hyponym*$_{\text{WN}}$ is a set of 2564 labelled pairs of nouns constructed in the following way. Pairs $\langle A, B \rangle$ were found in the list of nouns where $B$ is an ancestor of $A$ (i.e., $A$ lexically entails $B$). Each found pair is added either as a positive or a negative in the ratio 2:1 provided that the reverse pairing has not already been added and provided that each word has not previously been used in that position. Co-hyponym pairs (i.e., words which share a direct hypernym) were also found within the list of nouns. Each found pair is added to the data set (as a negative) provided the reverse pairing has not already been added, and provided that neither word has already been seen in that position in a pairing (either in the entailment pairs or the co-hyponym pairs). The same number of co-hyponym pairs as hypernym-hyponym negatives is selected. This provides a balanced data set where half of the pairs are positive examples of entailment and the other half are semantically similar but not entailing.

*cohyponym*$_{\text{WN}}$ is a set of 3771 labelled pairs of nouns. It was constructed in the same way as *hyponym*$_{\text{WN}}$ except the same number of co-hyponym pairs were selected as the total number of entailment pairs (in either direction). These co-hyponym pairs were labelled as positive and the entailment pairs were labelled as negative. Thus, this provides a balanced data set where half of the pairs are positive examples of co-hyponyms and the other half, the negative examples, are entailment pairs (with direction unspecified)

In both these sets, the average path distance between entailment pairs is 1.64, whereas path distance between co-hyponym pairs is 2.

### 3.4 Experimental Setup

Most of our experiments were carried out using an implementation of five-fold cross-validation using each combination of data set, vector set and classifier. In this setup, the pairs are randomly partitioned into five subsets, one subset is held out for testing whilst the classifiers are trained on the remaining four, and this process is repeated using each subset as the test set.

In initial experiments with the BLESS datasets, the SVM classifiers were able to achieve classification accuracy of over 95% for *hyponym*$_{\text{BLESS}}$ and over 90% for *cohyponym*$_{\text{BLESS}}$. However, the results using random vectors were not significantly different from using the distributional vectors harvested from Wikipedia. This indicated that the classifiers were learning ontological information implicit in the training data. In order to address this, when using the BLESS datasets, we removed any pair from the training data if either word was present in the test data. In order to preserve a reasonable amount of training data, we implemented this approach with ten-fold cross-validation. In all subsequent experiments, across all datasets and classifiers, we found performance by the random vectors was no higher than 52%. This indicates that the performance seen in Table 3 is due to learning from distributional features rather than any ontological information implicit in the training set.

## 4 Results

In Table 3, we compare average accuracy for a number of different classifiers on each of two tasks, distinguishing hyponyms and distinguishing co-hyponyms, on each of the two datasets.

Looking at the results for the *hyponym*$_{\text{BLESS}}$ data set, we can see that the SVM methods do generally outperform the unsupervised methods. However, the best performing model is svmSING, suggesting that, for this data set, it is best to try to learn the distributional features of more general terms, rather than comparing the vector representations of the two terms under consideration.

[6]Note that imposing these requirements on the BLESS data sets would lead to very small data sets, since information is only provided for 200 nouns.

| dataset | svmDIFF | svmMULT | svmADD | svmCAT | svmSING | knnDIFF | | |
|---|---|---|---|---|---|---|---|---|
| $hyponym_{\text{BLESS}}$ | **0.74** | 0.56 | 0.66 | 0.68 | **0.75** | 0.54 | | |
| $cohyponym_{\text{BLESS}}$ | 0.62 | 0.39 | 0.41 | 0.40 | 0.40 | 0.58 | | |
| $hyponym_{\text{WN}}$ | **0.75** | 0.45 | 0.37 | **0.74** | 0.69 | 0.50 | | |
| $cohyponym_{\text{WN}}$ | 0.37 | 0.60 | **0.68** | **0.64** | 0.58 | 0.50 | | |
| dataset | most freq | cosineP | linP | widthdiff | singlewidth | CRdiff | invCLP | balAPincP |
| $hyponym_{\text{BLESS}}$ | 0.54 | 0.53 | 0.54 | 0.56 | 0.58 | 0.52 | 0.54 | 0.54 |
| $cohyponym_{\text{BLESS}}$ | 0.61 | **0.79** | **0.78** | - | - | - | - | - |
| $hyponym_{\text{WN}}$ | 0.50 | 0.53 | 0.52 | 0.70 | 0.65 | 0.70 | 0.66 | 0.53 |
| $cohyponym_{\text{WN}}$ | 0.50 | 0.50 | 0.55 | - | - | - | - | - |

Table 3: Accuracy Figures for the data sets generated from BLESS and WordNet (standard errors $<$ 0.02). For cohyponyms, results for measures designed to detect hyponymy have been omitted. We also omit results of clarkediff as these were consistently the same or less than CRdiff.

On the corresponding co-hyponym task, using the $cohyponym_{\text{BLESS}}$ data set, we see the best performing classifier is the cosine measure. The cosine measure is able to perform relatively well here because a substantial proportion of the negative examples (25%) are random unrelated words which will have low cosine scores. It is also consistent with earlier work (e.g., (Lenci and Benotto, 2012)) which suggests that measures such as the cosine measure "prefer" words in symmetric semantic relationships such as co-hyponymy. The poor performance of the SVM methods here can perhaps be explained by the paucity of the training data in this experimental set up with this data set. If, for example, our test concept is `robin`, our approach requires that we will not have any training pairs containing `robin`, or any training pairs containing any of the words to which `robin` is related in the test set. In a dataset as small as BLESS, this requirement effectively removes all knowledge of the distributional features of words in the target domain. Hence, the need for a larger dataset as we have extracted from WordNet.

Looking at the results for the $hyponym_{\text{WN}}$ data set, the directional SVM methods (*svmDIFF* and *svmCAT*) substantially outperform the symmetric SVM methods, and their performance is significantly better (at the 0.01% level) than the unsupervised methods. Also of note is the substantial difference between *svmDIFF* and *knnDIFF*. Both of these methods are trained on the differences of vectors. However, the linear SVM outperforms kNN by 19–25%. This may suggest that the shape of the vector space inhabited by the positive entailment pairs is particularly conducive for learning a linear SVM. Positive and negative pairs are close together (as evidenced by the poor performance of *kNN*), but generally linearly separable.

Looking at the results for the $cohyponym_{\text{WN}}$ data set, it is clear that the unsupervised methods cannot distinguish the co-hyponym pairs from the entailing pairs. The supervised SVM methods do substantially better, with the best performance achieved by *svmADD* and *svmCAT*. Both of these methods essentially retain information about all of the features of both words. *svmMULT* does much better than *svmDIFF*, which suggests that the shared features are more indicative than the non-shared features for this task.

The reasonably high performance of *svmSING* on both data sets suggests that words which have co-hyponyms in the data set tend to inhabit a somewhat different part of the feature space to words which are included as entailed words in the data set. We hypothesise that there are specific features which more general words tend to share (regardless of their topic) which makes it possible to identify more general words from more specific words. This is completely consistent with very recent results using SLQS, a new entropy-based measure (Santus et al., 2014). Here, the authors hypothesise that the most typical contexts of a hypernym are less informative than the most typical linguistic contexts of its hyponyms, with some promising results. It would be plausible to hypothesise that *svmSING* is learning which nouns typically have less informative contexts and are therefore likely to by hypernyms.

Given prior work, the performance of the *balAPincP* measure is lower than expected on the $hyponym_{\text{WN}}$ dataset. Our task is slightly different to that of (Kotlerman et al., 2010), since we are determining the existence (or not) of hyponymy, rather than the direction of entailment for pairs where it is known that a relationship exists. It could be that the measure is particularly suited to the latter task.

## 5 Conclusions and Further Work

We have shown that it is possible to predict to a large extent whether or not there is a specific semantic relation between two words given their distributional vectors, using a supervised approach based on linear SVMs. The increase in accuracy over unsupervised methods is significant at the 0.01% level and corresponds to a substantial absolute reduction in error rate (over 15%).

We have also shown that the choice of vector operation is significant. Whilst concatenating the vectors, and therefore retaining all of the information from both vectors including direction, generally performs well, we have also shown that different vector operations are useful in establishing different relationships. In particular, the vector difference operation, which loses information about the original vectors, achieved performance indistinguishable from concatenation on the entailment task, where the classifier is required to distinguish hyponyms from other semantically related words including hypernyms. On the other hand, the addition operation, which also loses information, outperformed concatenation by 4% (which is statistically significant at the 0.01% level) on the coordinate task, where the classifier is required to distinguish co-hyponyms from hyponyms and hypernyms. Hence the nature of the relationship one is trying to establish between words determines the nature of the operation one should perform on their associated vectors.

We have also shown that it is possible to outperform state-of-the-art unsupervised methods even when a data set has been constructed without ontological information, and when target words have not previously been seen in that position of a relationship in the training data. Hence, we believe the supervised methods are learning characteristics of the underlying feature space which are generalisable to new words (inhabiting the same feature space).

In future work, we intend to apply this approach to the problem of labelling the distributional neighbours found for a given word with specific semantic relations. We also plan to investigate the use of bag-of-words (windowed) vectors instead of grammatical relations for this task.

Finally, we believe that the data sets constructed from WordNet, which we publish alongside this paper, can be used as a useful benchmark in evaluating future advances in this area, both for supervised and unsupervised methods.

## Acknowledgements

This work was funded by UK EPSRC project EP/IO37458/1 “A Unified Model of Compositional and Distributional Compositional Semantics: Theory and Applications”.

## References

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 workshop on Geometric Models of Natural Language Semantics, EMNLP 2011*.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32. Association for Computational Linguistics.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2010. Global learning of focused entailment graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1220–1229, Uppsala, Sweden, July. Association for Computational Linguistics.

Shane Bergsma, Aditya Bhargava, Hua He, and Grzegorz Kondrak. 2010. Predicting the semantic compositionality of prefix verbs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 293–303, Cambridge, MA, October. Association for Computational Linguistics.

Danushka Bollegala, David Weir, and John Carroll. 2011. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*.

Kenneth Ward Church and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics*, ACL '89, pages 76–83, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daoud Clarke. 2009. Context-theoretic semantics for natural language: an overview. In *Proceedings of the Workshop of Geometric Models for Natural Language Semantics*.

James Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

Georgiana Dinu and Stefan Thater. 2012. Saarland: Vector-based models of semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*.

Christaine Fellbaum, editor. 1989. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, Massachusetts.

Trevor Fountain and Mirella Lapata. 2012. Taxonomy induction using hierarchical random graphs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 466–476, Montréal, Canada, June.

Maayan Geffet and Ido Dagan. 2005. Lexical entailment and the distributional inclusion hypothesis. In *Proceedings of the 43rd meeting of the Association for Computational Liuguistics (ACL)*, pages 107–114.

Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. 2011. Concrete sentence spaces for compositional distributional models of meaning. *Proceedings of the 9th International Conference on Computational Semantics (IWCS 2011)*, pages 125–134.

Zelig Harris. 1954. Distributional structure. *Word*, 10:146–162.

Mitesh Khapra, Anup Kulkarni, Saurabh Sohoney, and Pushpak Bhattacharyya. 2010. All words domain adapted WSD: Finding a middle ground between supervision and unsupervision. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1532–1541, Uppsala, Sweden, July.

Adam Kilgarriff and Colin Yallop. 2000. What's in a thesaurus? In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC2000)*.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Special Issue of Natural Language Engineering on Distributional Lexical Semantics*, 4(16):359–389.

Lillian Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, College Park, Maryland, USA, June.

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*Sem)*.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING 1998)*.

Tara McIntosh. 2010. Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 356–365, Cambridge, MA, October. Association for Computational Linguistics.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June.

Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 2012. Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In *Proceedings of COLING 2012*, pages 1781–1796, Mumbai, India, December.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio, June. Association for Computational Linguistics.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the ACL workshop on Incremental Parsing*, pages 50–57.

Marek Rei and Ted Briscoe. 2013. Parser lexicalisation through self-learning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 391–400, Atlanta, Georgia, June. Association for Computational Linguistics.

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

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*.

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*, pages 801–808. Association for Computational Linguistics.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211.

György Szarvas, Chris Biemann, and Iryna Gurevych. 2013. Supervised all-words lexical substitution using delexicalized features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1131–1141, Atlanta, Georgia, June.

Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 849–856, Manchester, UK, August.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 81–88.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of Coling 2004*, pages 1015–1021, Geneva, Switzerland, Aug 23–Aug 27. COLING.

Julie Weeds, David Weir, and Jeremy Reffin. 2014. Distributional composition using higher-order dependency vectors. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality, EACL 2014*, Gothenburg, Sweden, April.

Dominic Widdows. 2008. Semantic vector products: Some initial investigations. In *Proceedings of the Second Symposium on Quantum Interaction, Oxford, UK*, pages 1–8.

Wen-tau Yih, Geoffrey Zweig, and John 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*, pages 1212–1222, Jeju Island, Korea, July. Association for Computational Linguistics.

Chen Zhang and Joyce Chai. 2010. Towards conversation entailment: An empirical investigation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 756–766, Cambridge, MA, October. Association for Computational Linguistics.