# Inclusive yet Selective: Supervised Distributional Hypernymy Detection

**Stephen Roller***, **Katrin Erk**†, **Gemma Boleda**†
* Department of Computer Science
† Department of Linguistics
The University of Texas at Austin
roller@cs.utexas.edu, katrin.erk@mail.utexas.edu,
gemma.boleda@upf.edu

## Abstract

We test the Distributional Inclusion Hypothesis, which states that hypernyms tend to occur in a superset of contexts in which their hyponyms are found. We find that this hypothesis only holds when it is applied to relevant dimensions. We propose a robust supervised approach that achieves accuracies of .84 and .85 on two existing datasets and that can be interpreted as selecting the dimensions that are relevant for distributional inclusion.

## 1 Introduction

One of the main criticisms of distributional models has been that they fail to distinguish between semantic relations: Typical nearest neighbors of *dog* are words like *cat*, *animal*, *puppy*, *tail*, or *owner*, all obviously related to *dog*, but through very different types of semantic relations. On these grounds, Murphy (2002) argues that distributional models cannot be a valid model of conceptual representation. Distinguishing semantic relations are also crucial for drawing inferences from distributional data, as different semantic relations lead to different inference rules (Lenci, 2008). This is of practical import for tasks such as Recognizing Textual Entailment or RTE (Geffet and Dagan, 2004).

For these reasons, research has in recent years started to attempt the detection of specific semantic relationships, and current results suggest that distributional models can, in fact, distinguish between semantic relations, given the right similarity measures (Weeds et al., 2004; Kotlerman et al., 2010; Lenci and Benotto, 2012; Herbelot and Ganesalingam, 2013; Santus, 2013). Because of its relevance for RTE and other tasks, much of this work has focused on hypernymy. Hypernymy is the semantic relation between a superordinate term in a taxonomy (e.g. *animal*) and a subordinate term (e.g. *dog*).

Distributional approaches to date for detecting hypernymy, and the related but broader relation of lexical entailment, have been unsupervised (except for Baroni et al. (2012)) and have mostly been based on the Distributional Inclusion Hypothesis (Zhitomirsky-Geffet and Dagan, 2005; Zhitomirsky-Geffet and Dagan, 2009), which states that more specific terms appear in a subset of the distributional contexts in which more general terms appear. So, *animal* can occur in all the contexts in which *dog* can occur, plus some contexts in which *dog* cannot – for instance, *rights* can be a typical cooccurrence for *animal* (e.g. "animal rights"), but not so much for *dog* (e.g. #"dog rights").

This paper takes a closer look at the Distributional Inclusion Hypothesis for hypernymy detection. We show that the current best unsupervised approach is brittle in that their performance depends on the space they are applied to. This raises the question of whether the Distributional Inclusion Hypothesis is correct, and if so, under what circumstances it holds. We use a simple supervised approach to relation detection that has good performance (accuracy .84 on BLESS, .85 on the lexical entailment dataset of Baroni et al. (2012)) and works well across different spaces.[1] Furthermore, we show that it can be interpreted as selecting dimensions for which the Distributional Inclusion Hypothesis does hold. So, our answer is to propose the *Selective Distributional Inclusion Hypothesis*: The Distributional Inclusion Hypothesis holds, but only for relevant dimensions.



[1] Code and data are available at http://stephenroller.com/research/coling14.



*Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland, August 23-29 2014.

## 2 Background

**Distributional models.** Distributional models represent a word through the contexts in which it has been observed, usually in the form of a vector representation (Turney and Pantel, 2010). A target word is represented as a vector in a high-dimensional *space* in which the *dimensions* are context items (for example, other words) and the coordinates of the vector indicate the target's degree of association with each context item. In this paper, we also use dimensionality reduced spaces in which dimensions do not stand for individual context items anymore.

**Pattern-based approaches to inducing semantic relations.** Early work on automatically inducing semantic relations between words, starting with Hearst (1992), uses textual patterns. For example, "[$NP_1$] and other [$NP_2$]" implies that $NP_2$ is a hypernym of $NP_1$. Pattern-based approaches have been applied to meronymy (Berland and Charniak, 1999; Girju et al., 2003; Girju et al., 2006), synonymy (Lin et al., 2003), co-hyponymy (Snow et al., 2005), hypernymy (Cimiano et al., 2005), and several relations between verbs (Chklovski and Pantel, 2004). Pantel and Pennachiotti (2006) generalize the idea to a wide variety of relations. Turney (2006) uses vectors of patterns to determine similarity of semantic relations. A task related to semantic relation induction is the extension of an existing taxonomy (Buitelaar et al., 2005). Snow et al. (2006) do this by using hypernymy and co-hyponymy detectors.

**Lexical entailment, hypernymy, and the Distributional Inclusion Hypothesis.** Weeds et al. (2004) introduce the notion of *distributional generality*, where $v$ is distributionally more general than $u$ if $u$ appears in a subset of the contexts in which $v$ is found, and speculate that hypernyms ($v$) should be more distributionally general than hyponyms ($u$). Zhitomirsky-Geffet and Dagan (2005; 2009) introduce the term *Distributional Inclusion Hypothesis* for the idea that distributional generality encodes hypernymy or the more loosely defined relation of *lexical entailment*.

Weeds and Weir (2003) measure distributional generality using a notion of precision (eq. 1). Here and in all equations below, $u$ is the narrower term, and $v$ the more general one. Abusing notation, we write $u$ for both a word and its associated vector $\langle u_1, \ldots, u_n \rangle$. Kotlerman et al. (2010) predict lexical entailment with the *balAPinc* measure, a modification of the Average Precision (AP) measure (eq. 2). The general notion is that scores should increase with the number of dimensions of $v$ that $u$ shares, and also give more weight to the highly ranked dimensions (i.e. largest magnitude) of the narrower term $u$. This is captured in *APinc* by computing precision $P(r)$ at every rank $r$ among $u$'s dimensions – where precision is the fraction of dimensions shared with $v$ –, and weighting by the rank of the same dimension in the broader term, $rel'(v, r, u)$. The final measure, *balAPinc*, smooths using the *LIN* similarity measure (Lin, 1998). (We only sketch this measure here due to its complexity; details are given in Kotlerman et al. (2010).)

$$1(x) = \begin{cases} 1 \text{ if } x > 0; \\ 0 \text{ otherwise} \end{cases}$$

$$WeedsPrec(u, v) = \frac{\sum_{i=1}^{n} u_i \cdot 1(v_i)}{\sum_{i=1}^{n} u_i} \tag{1}$$

$$\text{APinc}(u, v) = \frac{\sum_{r=1}^{|1(u)|} P(r) \cdot rel'(v, r, u))}{|1(u)|} \tag{2}$$

$$\text{balAPinc}(u, v) = \sqrt{\text{APinc}(u, v) \cdot \text{LIN}(u, v)}$$

The *ClarkeDE* measure (Clarke, 2009) computes degree of entailment as the degree to which the narrower term $u$ has lower values than $v$ across all dimensions (eq. 3). Lenci and Benotto (2012) introduce the *invCL* measure, which uses *ClarkeDE* to measure both distributional inclusion of $u$ in $v$ and distributional *non-inclusion* of $v$ in $u$ (eq. 4). While all other measures interpret the Distributional Inclusion Hypothesis as the degree to which a $\subseteq$ relation holds, Lenci and Benotto test the degree to which proper inclusion $\subsetneq$ holds. They consider not only the degree to which the contexts of the narrower terms are included in the contexts of the wider term, but also determine the degree to which the wider term has contexts that the narrower term does not have.

$$\text{CL}(u, v) = \frac{\sum_{i=1}^{n} \min(u_i, v_i)}{\sum_{i=1}^{n} u_i} \tag{3}$$

$$\text{invCL}(u, v) = \sqrt{\text{CL}(u, v) \cdot (1 - \text{CL}(v, u))} \tag{4}$$

Like Lenci and Benotto, we focus on the stricter hypernymy relation, rather than lexical entailment. We believe that the different relations that make up lexical entailment have different distributional indications and that, for that reason, it will be easier to detect the relations separately than together.

Baroni et al. (2012) proposes a supervised approach to hypernymy detection that represents two words as the concatenation of their vectors. They also mention in passing another supervised approach that represents two words as the component-wise difference of their vectors. These are broadly the two approaches that we test, though we introduce significant modifications.

## 3 Data

### 3.1 Distributional Vector Spaces

We use three standard types of distributional spaces.

**U+W2:** This space is based on a concatenation of the Gigaword, BNC, English Wackypedia and ukWaC corpora (Baroni et al., 2009). The corpora are POS-tagged and lemmatized. We keep only content words (nouns, proper nouns, adjectives and verbs) with a corpus frequency of 500 or larger. The resulting U+ corpus has roughly 133K word types and 2.8B word tokens. We created a vector space by counting co-occurrences of these word types within a window of two words on the left and the right, using the top 20k most frequent content words as dimensions. The space was transformed using Positive Pointwise Mutual Information (PPMI).

**U+Sent:** The U+Sent space is constructed the same way as U+W2, but uses full sentence contexts instead of 2-word windows.

**TypeDM:** This space is extracted from the TypeDM tensors (Baroni and Lenci, 2011). TypeDM contains a list of weighted tuples, $\langle\langle w_1, l, w_2\rangle, \sigma\rangle$, where $w_1$ and $w_2$ are content words, $l$ is a corpus-derived syntagmatic relationship between the words, and $\sigma$ is a weight estimating saliency of the relationship. We construct vectors for every unique $w_1$ using the set of $\langle l, w_2\rangle$ pairs as dimensions and corresponding $\sigma$ values as dimension weights. We select TypeDM for its excellent performance in previous comparisons of distributional hypernymy measures (Lenci and Benotto, 2012).

**Reduced Spaces:** In some experiments, we use dimensionality reduced spaces. We reduce all three spaces to 300 dimensions using Singular Value Decomposition. We use a subscript to denote reduced spaces, e.g. U+W2$_{300}$. When necessary, we use the term *original dimensions* to refer to the vector dimensions from the original, non-reduced spaces (e.g. U+W2); the term *latent dimensions* refers to the dimensions in the reduced spaces (e.g. U+W2$_{300}$).

### 3.2 Evaluation Data Sets

**BLESS:** The BLESS data set (Baroni and Lenci, 2011) covers 200 *concepts*, or concrete and unambiguous terms (divided into 17 different general *concept classes*, including *vehicle* and *ground mammal*), and their relationships to other nouns, called *relata*. Example concepts include *van* and *horse*. Each concept is related to several relata through different *semantic relations*. Following Lenci and Benotto (2012), we focus on the four semantic relations where both concepts and relata are nouns, for a total 14K data points: Hypernymy, denoting a superset relationship (e.g. *animal-dog*); Co-hyponymy, denoting words that share a common hypernym (e.g. *dog-cat*); Meronymy, denoting a part-whole relationship (e.g. *tail-dog*); and Random, denoting no relationship between the words (e.g. *dog-computer*).

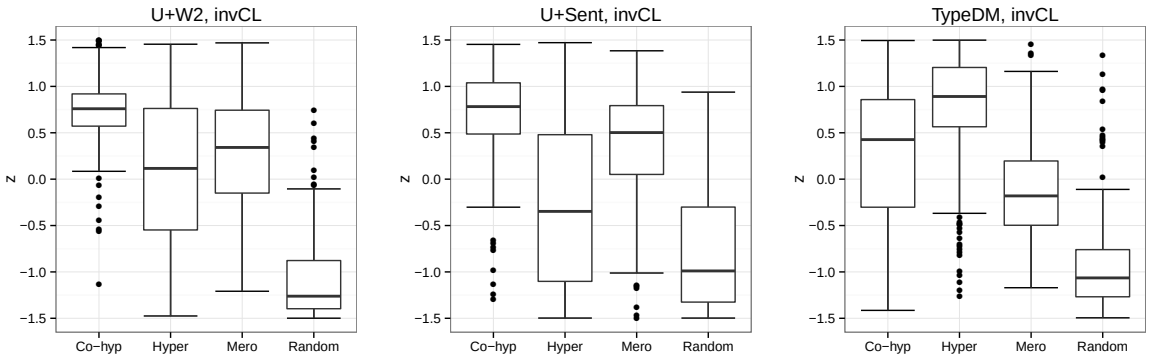


Figure 1: Distributions of relata *invCL* scores for the U+W2, U+Sent, and TypeDM spaces for each of the semantic relations, after per-concept $z$-normalization.

**ENTAILMENT:** (Baroni et al., 2012): The ENTAILMENT data set consists of 2,770 word pairs, balanced between positive (*house-building*) and negative (*leader-rider*) examples of hypernymy, with 1376 unique hyponyms and 1016 unique hypernyms. The positive examples were generated by selecting direct hypernym relationships from WordNet, the negative examples by randomly permuting the hypernyms of the positive examples, and then manually checking correctness.

## 4 Distributional Inclusion across Spaces

We test several unsupervised distributional approaches to hypernymy detection from the literature, focusing on the underlying vector space representation as the main parameter that we vary. We use the three spaces described in Section 3. We test four hypernymy detection approaches, all of them similarity measures based on the Distributional Inclusion Hypothesis: *WeedsPrec*, *balAPinc*, *ClarkeDE*, and *invCL*. Our baseline is the standard *cosine* measure. We evaluate on the BLESS dataset.

To evaluate on BLESS, we follow the evaluation scheme laid out in Baroni and Lenci (2011). Given a space and similarity measure, we compute similarity for each concept and relatum. For each concept, we select its nearest neighbors (according to the given similarity measure) in each of the four relations (CO-HYP, HYPER, MERO, RANDOM), and transform the corresponding four similarities to $z$-scores. Across all concepts, this yields four sets of z-normalized similarity scores, one for each relation. These four sets describe the relative similarity of concepts to their nearest neighbors in different relations. Tukey's Honestly Significant Difference test is used for testing whether scores differ significantly between relations (threshold: $p < 0.05$).

Figure 1 shows the distributions of $z$-scores for *invCL* for the four relations, with one graph for each of the three spaces we consider. For this illustration, we focus on *invCL* because it shows the overall best performance at identifying hypernymy. The rightmost plot in Figure 1 replicates the analysis of Lenci and Benotto (2012), who used the TypeDM space. It confirms their finding that *invCL* gives significantly higher values to hypernyms than co-hyponyms – at least on this space. However, in the U+W2 and U+Sent spaces (leftmost and middle plot), *invCL* clearly loses any ability to rank hypernyms the highest; indeed, in both spaces, co-hyponymy and meronymy both have significantly higher $z$-scores than hypernymy. Concerning the other measures, we found that they patterned with *invCL*. On TypeDM, *ClarkeDE* and *WeedsPrec* had significantly higher nearest-neighbor values for hypernyms than co-hyponyms.[2] On U+W2 and U+Sent, all measures ranked co-hyponyms significantly higher than hypernyms. With the baseline measure, *cosine*, the similarity ratings for the CO-HYP relation are always the highest, no matter the space, followed by HYPER, MERO, RANDOM in this order.

Following Kotlerman et al. (2010) and Lenci and Benotto (2012), we also report the performance of the measures using Mean Average Precision (MAP). Average Precision (AP) is a measure often used in

[2] *balAPinc* could not be evaluated on TypeDM due to computational issues.

| Measure | CO-HYP | HYPER | MERO | RANDOM |
|---|---|---|---|---|
| U+W2 | | | | |
| *cosine* | .68 | .20 | .27 | .27 |
| *ClarkeDE* | .66 | .19 | .28 | .28 |
| *invCL* | .60 | .18 | .31 | .28 |
| U+Sent | | | | |
| *cosine* | .66 | .18 | .28 | .28 |
| *ClarkeDE* | .66 | .15 | .29 | .28 |
| *invCL* | .59 | .13 | .34 | .29 |
| TypeDM | | | | |
| *cosine* | .78 | .19 | .20 | .29 |
| *ClarkeDE* | .45 | .35 | .25 | .32 |
| *invCL* | .38 | .36 | .27 | .33 |

Table 1: Mean Average Precision for the unsupervised measures on three spaces.

the Information Retrieval community with a maximal AP score of 1 when all relevant documents (relata with the right relationship, in our case) are ranked at the top. We compute AP on a per-concept basis and report the mean over all 200 AP values. An advantage of MAP is that, while the BLESS analysis method focuses on nearest neighbors, MAP evaluates the ranking of all relata. A disadvantage of MAP is that it does not test the degree to which a similarity measure separates different semantic relations, like Tukey does, so it may overstate the discriminative power of a particular measure. However, it provides a more intuitive accuracy-like number compared to the BLESS evaluation.

Table 1 shows the Mean Average Precision values for *cosine*, *ClarkeDE*, and *invCL* on all three spaces. We also computed *WeedsPrec* and *balAPinc* results, obtaining the same picture; we focus on *ClarkeDE* and *invCL* because *ClarkeDE* is a component of *invCL*, and *invCL* is the current best measure. The results corresponding to Lenci and Benotto's are shown in the lowest part of Table 1, where we report numbers for TypeDM. Like Lenci and Benotto, we find that unsupervised measures other than *invCL* rank co-hyponyms the highest, and obtain relatively low results for hypernyms. For *invCL* in TypeDM, Lenci and Benotto obtain 0.38 MAP for co-hyponyms and a slightly higher 0.40 for hypernyms, though they do not report significance testing results. We obtain 0.38 for co-hyponyms and 0.36 for hypernyms, and the difference is not significant.[3] Even though our results are slightly different from those in Lenci and Benotto (2012), both our results and theirs point to at most a weak preference of *invCL* for hypernyms over co-hyponyms. Moreover, in the U+W2 and U+Sent spaces we see that all three measures are very poor at identifying hypernyms, and the co-hyponymy relation stubbornly persists as most relevant to all three measures, by a large margin.

Our results thus constitute a puzzle for the Distributional Inclusion Hypothesis. It seems that there must be some merit to the hypothesis: On one particular space, namely TypeDM, the nearest neighbors in the hypernymy relation had higher similarity scores than any other relation by a significant margin. This was true for all the hypernymy detectors we studied. But even on TypeDM, the MAP evaluation showed at most a weak hypernymy signal, and when spaces other than TypeDM were used, the effect vanished altogether. So how strong an indication for hypernymy can we expect from distributional inclusion measures in general? We will return to this question below, where our answer will be: The Distributional Inclusion Hypothesis seems to hold after all, but it needs to be applied to the right kind of dimensions – and a supervised approach can help in picking the right dimensions.

As the unsupervised approaches struggle to detect hypernymy and do not seem robust to changes in standard space parameters, we think it is time to consider supervised approaches. In the next section, we explore two simple supervised approaches that show good performance and are robust to changes in the underlying space.

[3] Wilcoxon signed-rank test.

## 5 Supervised Hypernymy Detection

We use two simple, supervised models for predicting BLESS and ENTAILMENT relations. The first (Concat) is a model previously proposed by Baroni et al. (2012). The second (Diff) takes up an idea from a footnote in Baroni et al. (2012), but while that footnote stated that the approach in question did not work, we find that, with a few modifications, it obtains the best performance – and can be interpreted as a supervised version of the Distributional Inclusion Hypothesis. Note that while we used unreduced spaces in the previous section, we now use reduced spaces throughout (these are the spaces with the $_{300}$ subscript), in order not to have more features than data points.

### 5.1 Models, Features, and Method

**Concat:** We use a standard Support Vector Machine (SVM) classifier with a concatenation of vectors as input features. SVMs are binary classifiers which learn the maximum margin hyperplane separating the two classes. SVMs employ kernel functions to find the hyperplanes in higher dimensional spaces which are nonlinear in the original space. As feature vectors for the classifier, we follow Baroni et al. (2012) and use the concatenation of the latent dimension vectors representing words. For the ENTAILMENT dataset, we use the concatenation of the hyponym latent vector and the hypernym latent vector for each word pair as training features, and the *entails/doesn't entail* annotations as binary targets. For BLESS, we use the concatenation of the concept latent vector and the relatum latent vector as training features, and the four relationship classes as targets. We choose the four-way task rather than a "hypernymy vs. other" classification because BLESS contains many more co-hyponymy and random than hypernymy pairs, which would give a very high baseline in the two-way task. Additionally, the other relations in BLESS, in particular meronymy, may be interesting in their own right.

Since SVMs are binary classifiers, we use SciKit-Learn's default setting to train 6 pairwise-relation one-vs-one classifiers which vote on the final answer. We use a polynomial kernel with a degree of 3 and a penalty term of $C = 1.0$, and all other hyperparameters are chosen using the SciKit-Learn default values (Pedregosa et al., 2011). No hyperparameters are tuned in any experiment.

**Diff:** Our second classifier is a Logistic Regression (aka MaxEnt) model trained on difference vectors. Logistic Regression is a statistical model for binary classification. It learns a linear hyperplane separating the classes and estimates a probability for classes using a logistic function. We selected Logistic Regression over other possible linear classifiers for its natural ability to give likelihood estimates, which we believe will be useful in future work in an application of hypernymy classification to RTE.

As feature vectors, we use a Mikolov-inspired method of representing word pairs as the *difference vectors* between the two words.[4] Baroni et al. (2012) suggested the use of difference vectors as input to a classifier, but reported them as unsuccessful. We found difference vectors to be excellent features, with three important modifications: a linear classifier is better than a nonlinear one; vectors must be normalized to have a magnitude of 1 before taking the difference; and squared difference vectors must also be included as features. So, we represent each word pair with latent vectors $(u, v)$ as a two part vector $\langle f; g \rangle$, where

$$f_i = \frac{u_i}{\|u\|} - \frac{v_i}{\|v\|},$$
$$g_i = {f_i}^2.$$

These differences features[5] are analogous to a *supervised* distributional inclusion measure. The difference between two words on a particular dimension captures the degree of distributional inclusion on that dimension. The primary distinction between the difference features and the unsupervised measures is that the supervised classifier learns to weight the importance of different dimensions. The $f$ features encode directional aspects of distributional inclusion: that the hyponym contexts should be included in

[4] After recent work using subtraction to represent analogy in certain neural-network spaces (Mikolov et al., 2013).

[5] We also tried variations, such as not normalizing vectors and removing the difference squared vector, but found this setting the best. We also tried the Diff features with an SVM and other nonlinear classifiers, but they performed worse.

| Data set | BLESS | | ENTAILMENT | |
|---|---|---|---|---|
| Baseline | .46 | | .50 | |
| Classifier | Concat | Diff | Concat | Diff |
| U+W2$_{300}$ | .76 | **.84** | .81 | **.85** |
| U+Sent$_{300}$ | .73 | .80 | .78 | .82 |
| TypeDM$_{300}$ | - | .82 | .65 | **.85** |

Table 2: Average accuracy of Concat and Diff on BLESS and ENTAILMENT using different spaces for feature generation.

those of the hypernym (the weight learned is positive), and the hypernym contexts should not be included in those of the hyponym (the weight learned is negative). So like *invCL*, this model uses a "proper subset" interpretation of the Distributional Inclusion Hypothesis, but only considers selected dimensions (i.e. those that the model assigns nonzero weights).

The difference-squared features ($g$), on the other hand, typically identify dimensions that are *not* indicative of hypernymy, by learning negative weights on them (more about this in Section 6). Thus, rather than helping identify hypernyms, they help separate random relations from the rest.

We use a L1 regularizer with a strength of $C = 1.0$. All other hyperparameters are chosen using the SciKit-Learn defaults. Since Diff is also a binary classifier, we use SciKit-Learn's default setting of training 4 one-vs-all classifiers for BLESS, with the most confident classifier choosing the final answer.

**Method:** For evaluation on BLESS, we hold out one concept and train on the remaining 199 concepts. We also exclude from the training set any pair containing a relatum which appears in the test set. This way, no word that appears in the test set has been seen in training. We report the average accuracy across all concepts. We use the most frequent relation type (random) as our baseline. For the ENTAILMENT data set, we hold out one hyponym and train on all remaining hyponyms. Again, we exclude from training any pair containing a hypernym which appears in the test set. We report average accuracy across all hyponyms. The data set is balanced, so the baseline is 0.5.

### 5.2 Results

Table 2 shows the performance of the two classifiers, Concat and Diff, on both the BLESS and ENTAILMENT datasets, using three underlying spaces. We use the reduced versions of the three spaces, indicated by the subscript $_{300}$. Note that the Concat classifier could not converge using features from TypeDM$_{300}$, so we omit the result. With both methods, we obtain a high accuracy on the two datasets, with results around .8 against baselines around .5. Our best result is .84 on BLESS and .85 on ENTAILMENT. Moreover, both approaches are in general robust to changes in space parameters (with TypeDM/Concat an outlier). Still, the U+W2$_{300}$ space seems to be the best for this task: Its scores are significantly[6] higher than the rest, except for TypeDM on ENTAILMENT, which achieves the same score as U+W2$_{300}$. Diff achieves significantly higher results than Concat.

When provided more information, Concat outperforms Diff. For instance, if cross-validation is done over all pairs in BLESS in the U+W2$_{300}$ space, Concat achieves .98 accuracy, while Diff obtains .90. However, in this setting the same words appear in the training and test sets (albeit in different pairs). We take this to mean that Concat is memorizing, rather than learning the hypernymy relation. This emphasizes the need for our stricter evaluation that prevents repetition between training and test sets.

Clearly, both classifiers do fairly well at predicting hypernymy relations between words, regardless of space. Naturally, one should ask what are the classifiers capturing that the unsupervised measures are missing? We propose that the supervised classifiers perform essentially the same operation as the unsupervised measures, but are learning to determine the relevance of dimensions. In particular, Diff is learning weights on vector difference features. This is equivalent to doing selective distributional inclusion. In the next section, we test this Selective Distributional Inclusion Hypothesis.

[6]Wilcoxon signed-rank test, $p < .001$.

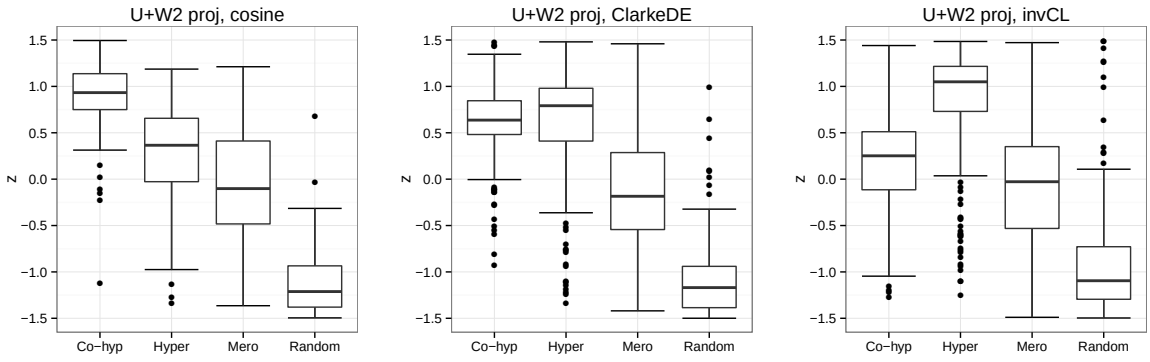


Figure 2: Distributions of relata scores across concepts using the *cosine*, *ClarkeDE*, and *invCL* measures (after per-concept z-normalization). Here we use the selected dimensions of the U+W2$_{proj}$ space.

## 6 Selective Distributional Inclusion

In order to test how well our supervised model is capturing the notion of selective distributional inclusion, we test each of the unsupervised measures on a smaller space, limited only to the dimensions preferred by the classifier. We emphasize that we do *not* aim to show that our supervised method outperforms unsupervised methods, but rather that the unsupervised methods benefit greatly from feature selection. Additionally, we analyze which dimensions are selected by the classifier to facilitate understanding of why these dimensions are important.

### 6.1 Experiment

We train the Diff classifier using the dimensionality-reduced U+W2$_{300}$ space with the same method we use in Section 5. We take the classifier's learned hyperplane separating hypernyms from other relations, and project the hyperplane back into the original U+W2 space.[7] We select the 500 dimensions in the original space that are most relevant according to the classifier weights, and test the unsupervised measures on this new space, which we denote as U+W2$_{proj}$.[8]

The 500 most relevant dimensions are selected as follows: We select the 250 most negatively weighted original dimensions using the difference features $f$. These are the features that have smaller values for hyponyms (e.g. *dog*) than for hypernyms (e.g. *animal*), so they characterize hypernymy. We further select the 250 most positively weighted original dimensions using the squared-differences features $g$. These are the ones where a large difference does not indicate hypernymy.

Figure 2 shows the boxplots for the BLESS analysis: the distributions of nearest-neighbor similarity scores for the four different semantic relations, for the measures *cosine*, *ClarkeDE*, and *invCL*. We see that *invCL* now easily discriminates hypernymy from the other relations in the backprojected space. (The difference of HYPER and CO-HYP is significant.) This is even though the space is based on U+W2, where *invCL* failed to rate hypernyms higher than co-hypernyms in Section 4. Unsurprisingly, *cosine*, which does not measure distributional inclusion, still prefers CO-HYP.

Table 3 shows the MAP scores for three of the measures in the new U+W2$_{proj}$ space. (The results for *balAPinc* and *WeedsPrec* are slightly worse than *ClarkeDE*.) All measures except for *cosine* assign higher scores to hypernyms than they did in the original space (compare to U+W2 part of Table 1). But it is only *invCL* that ranks hypernyms significantly higher than co-hyponyms.[9]

[7] Ideally we would train on the original space to inspect the relevant dimensions. However, there are more dimensions than examples, so we train on the SVD space and backproject.

[8] Note that U+W2$_{proj}$ varies slightly from concept to concept, since the hyperplane is learned on a per-concept basis. It is important that we use the linear Diff classifier for this reverse-projection procedure, as the separating hyperplane *must* be linear in order to complete the projection. In particular, the hyperplane in the Concat classifier cannot be easily backprojected, since it exists in a higher dimensional space than the projection matrix. Furthermore, it is important that we use a classifier trained using the difference features because of its analogy to the Distributional Inclusion Hypothesis.

[9] Wilcoxon signed-rank test, $p < .001$. To check that the measures are being improved by the dimension selection and not

| Measure | CO-HYP | HYPER | MERO | RANDOM |
|---|---|---|---|---|
| U+W2$_{proj}$ | | | | |
| *cosine* | .69 | .20 | .24 | .28 |
| *ClarkeDE* | .55 | .39 | .24 | .29 |
| *invCL* | .42 | **.58** | .24 | .29 |

Table 3: Mean Average Precision for the unsupervised measures after selecting the top dimensions from a supervised model.

For this experiment, we train on all of BLESS except for one concept and then evaluate the unsupervised models on the held-out concept – that is a setting that could, in principle, be used as a hypernymy detector. If we instead train the supervised model on all of BLESS to determine an upper bound of how well dimension selection can do on this dataset, MAP for *invCL* rises to .67.

Overall, these experiments provide strong evidence for the Selective Distributional Inclusion Hypothesis: The Distributional Inclusion Hypothesis holds, but only for relevant dimensions. In addition, hypernymy detectors need to test for "proper inclusion" of distributional contexts in order to really find hypernyms.

**Analysis of Selected Dimensions.** We examine the 500 dimensions selected by the above procedure, in order to see what the classifier is learning. As this is for analysis only, the dimensions were selected by training on all data.

Recall that the difference-squared $g$ features can be interpreted as dimensions that the classifier deems not indicative of hypernymy. 200 out of the 250 most relevant dimensions by $g$ are Computer Science related terms like *software*, *configure*, or *Linux*. Since ukWaC, the largest corpus we use, is web-based, it makes sense that it has many CS-related terms, which are noise when it comes to hypernymy detection for BLESS concepts. Also, we find that while the supervised approach needs the negative information from the $g$ features (for Diff in the U+W2$_{300}$ space, omitting $g$ features yields a drop from .84 to .8), the unsupervised measures cannot use it. Dropping $g$ features improves *invCL* results from .58 to .61. The $g$-based dimensions are explicitly those for which distributional inclusion should *not* hold, so they constitute noise to the unsupervised approaches.

The $f$ features can be interpreted as dimensions that characterize hypernyms. An inspection reveals two clear patterns. First, the features are topically relevant for the BLESS dataset. The 17 concept classes in the dataset belong to three broader groups: animals, plants, and artifacts. An annotation of the 250 dimensions by one of the authors showed that 58 dimensions are typical of animals (*parasite*, *extinct*), 14 typical of vegetables (*flora*, *nutrient*), 80 typical of artifacts (*repair*, *mechanical*), 49 are general terms (*find*, *worthy*), and 49 have no clear interpretation (*thee*, *enigmatic*). Second, the features are general terms. For instance, for animals we find terms like *animal*, *insect*, *creature*, *fauna*, *species*, *evolutionary*, *pathogen*, *nature*, *ecology*. We also find many hypernyms, including many concept class names.

Clearly, the selected features are domain dependent; most are directly related to the concepts and concept classes of BLESS. We expect that our method should work well for other data sets, given its high accuracy and the strict training procedure. However, these features are unlikely to be global indicators of hypernymy. This emphasizes the need, in future work, to find a way to automatically determine relevance on a per-word basis.

## 7 Conclusion

In this paper, we have tested the Distributional Inclusion Hypothesis, the basis for distributional approaches to hypernymy. We have found that the hypothesis only works if inclusion is selectively applied to a set of relevant dimensions.

just by restricting to a smaller space, we evaluated the similarity measures on a variation of the U+W2 space which uses 500 randomly selected dimensions from the original space. The results are approximately unchanged from those on the original U+W2 space.

We have tested two simple supervised approaches to distributional hypernymy detection and have found that they show good performance, and are robust to changes in the underlying space. Our best classifier achieves .84 accuracy on BLESS and .85 on the ENTAILMENT dataset of Baroni et al. (2012). It uses features that encode dimension-wise difference between vectors. This classifier can be interpreted as selecting the dimensions necessary for the Distributional Inclusion Hypothesis to work, thus as an effective way to implement *selective* distributional inclusion.

The next natural step is to use the supervised features to guide development of an unsupervised measure for hypernymy detection: Now that we have examples, we hope to propose a method which selects relevant features automatically. We also would like to explore detection of other relationships, such as meronymy. Finally, we would like to perform an extrinsic evaluation of our hypernymy detection approach in an actual RTE system.

## Acknowledgements

This research was supported by the DARPA DEFT program under AFRL grant FA8750-13-2-0026. The authors acknowledge the Texas Advanced Computing Center (TACC)[10] for providing grid resources that have contributed to these results. We thank the anonymous reviewers and the UTexas NLP group for their helpful comments and suggestions.

## References

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK, July. Association for Computational Linguistics.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

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, Avignon, France, April. Association for Computational Linguistics.

Matthew Berland and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 57–64, College Park, Maryland, USA, June. Association for Computational Linguistics.

Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. 2005. *Ontology Learning from Text: Methods, Evaluation and Applications*. Frontiers in Artificial Intelligence and Applications Series. IOS Press, Amsterdam.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 33–40.

Philipp Cimiano, Aleksander Pivk, Lars Schmidt-Thieme, and Steffen Staab. 2005. Learning taxonomic relations from heterogeneous sources of evidence. *Ontology Learning from Text: Methods, evaluation and applications*.

Daoud Clarke. 2009. Context-theoretic semantics for natural language: an overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 112–119, Athens, Greece, March. Association for Computational Linguistics.

Maayan Geffet and Ido Dagan. 2004. Feature vector quality and distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics*, page 247. Association for Computational Linguistics.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 1–8. Association for Computational Linguistics.

[10] `http://www.tacc.utexas.edu`

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2006. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, pages 539–545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aurélie Herbelot and Mohan Ganesalingam. 2013. Measuring semantic content in distributional vectors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 440–445, Sofia, Bulgaria, August. Association for Computational Linguistics.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16:359–389, 10.

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Alessandro Lenci. 2008. Distributional approaches in linguistic and cognitive research. *Italian Journal of Linguistics*, 20(1):1–31.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the 18th international Joint Conference on Artificial intelligence*, pages 1492–1493.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, volume 98, pages 296–304.

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. Association for Computational Linguistics.

Gregory L. Murphy. 2002. *The Big Book of Concepts*. MIT Press, Boston, MA.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertran Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, MMatthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Enrico Santus. 2013. SLQS: An entropy measure. Master's thesis, University of Pisa.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304, Cambridge, MA. MIT Press.

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*, ACL-44, pages 801–808, Stroudsburg, PA, USA. Association for Computational Linguistics.

Peter Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

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 the 20th International Conference on Computational Linguistics*, pages 1015–1021, Geneva, Switzerland, Aug 23–Aug 27. Association for Computational Linguistics, COLING.

Maayan Zhitomirsky-Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 107–114, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Maayan Zhitomirsky-Geffet and Ido Dagan. 2009. Bootstrapping distributional feature vector quality. *Computational linguistics*, 35(3):435–461.