Hierarchical Embeddings for Hypernymy Detection and Directionality

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, Ngoc Thang Vu
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5B, 70569 Stuttgart, Germany
{nguyenkh,koepermn,schulte,thangvu}@ims.uni-stuttgart.de

Abstract

We present a novel neural model *HyperVec* to learn hierarchical embeddings for hypernymy detection and directionality. While previous embeddings have shown limitations on prototypical hypernyms, *HyperVec* represents an unsupervised measure where embeddings are learned in a specific order and capture the hypernym–hyponym distributional hierarchy. Moreover, our model is able to generalize over unseen hypernymy pairs, when using only small sets of training data, and by mapping to other languages. Results on benchmark datasets show that *HyperVec* outperforms both state-of-the-art unsupervised measures and embedding models on hypernymy detection and directionality, and on predicting graded lexical entailment.

1 Introduction

Hypernymy represents a major semantic relation and a key organization principle of semantic memory (Miller and Fellbaum, 1991; Murphy, 2002). It is an asymmetric relation between two terms, a hypernym (superordinate) and a hyponym (subordiate), as in *animal–bird* and *flower–rose*, where the hyponym necessarily implies the hypernym, but not vice versa. From a computational point of view, automatic hypernymy detection is useful for NLP tasks such as taxonomy creation (Snow et al., 2006; Navigli et al., 2011), recognizing textual entailment (Dagan et al., 2013), and text generation (Biran and McKeown, 2013), among many others.

Two families of approaches to identify and discriminate hypernyms are predominent in NLP, both of them relying on word vector representations. ***Distributional count approaches*** make use of either directionally unsupervised measures or of supervised classification methods. Unsupervised measures exploit the *distributional inclusion hypothesis* (Geffet and Dagan, 2005; Zhitomirsky-Geffet and Dagan, 2009), or the *distributional informativeness hypothesis* (Santus et al., 2014; Rimell, 2014). These measures assign scores to semantic relation pairs, and hypernymy scores are expected to be higher than those of other relation pairs. Typically, Average Precision (AP) (Kotlerman et al., 2010) is applied to rank and distinguish between the predicted relations. Supervised classification methods represent each pair of words as a single vector, by using the concatenation or the element-wise difference of their vectors (Baroni et al., 2012; Roller et al., 2014; Weeds et al., 2014). The resulting vector is fed into a Support Vector Machine (SVM) or into Logistic Regression (LR), to predict hypernymy. Across approaches, Shwartz et al. (2017) demonstrated that there is no single unsupervised measure which consistently deals well with discriminating hypernymy from other semantic relations. Furthermore, Levy et al. (2015) showed that supervised methods memorize *prototypical hypernyms* instead of *learning* a relation between two words.

Approaches of hypernymy-specific embeddings utilize neural models to learn vector representations for hypernymy. Yu et al. (2015) proposed a supervised method to learn term embeddings for hypernymy identification, based on pre-extracted hypernymy pairs. Recently, Tuan et al. (2016) proposed a dynamic weighting neural model to learn term embeddings in which the model encodes not only the information of hypernyms vs. hyponyms, but also their contextual information. The performance of this family of models is typically evaluated by using an SVM to discriminate hypernymy from other relations.

In this paper, we propose a novel neural model *HyperVec* to learn hierarchical embeddings that **(i)** discriminate hypernymy from other relations (**detection task**), and **(ii)** distinguish between the hypernym and the hyponym in a given hypernymy relation pair (**directionality task**). Our model learns to strengthen the distributional similarity of hypernym pairs in comparison to other relation pairs, by moving hyponym and hypernym vectors close to each other. In addition, we generate a distributional hierarchy between hyponyms and hypernyms. Relying on these two new aspects of hypernymy distributions, the similarity of hypernym pairs receives higher scores than the similarity of other relation pairs; and the distributional hierarchy of hyponyms and hypernyms indicates the directionality of hypernymy.

Our model is inspired by the *distributional inclusion hypothesis*, that prominent context words of hyponyms are expected to appear in a subset of the hypernym contexts. We assume that each context word which appears with both a hyponym and its hypernym can be used as an indicator to determine which of the two words is semantically more general: Common context word vectors which represent distinctive characteristics of a hyponym are expected to be closer to the hyponym vector than to its hypernym vector. For example, the context word *flap* is more characteristic for a *bird* than for its hypernym *animal*; hence, the vector of *flap* should be closer to the vector of *bird* than to the vector of *animal*.

We evaluate our *HyperVec* model on both unsupervised and supervised hypernymy detection and directionality tasks. In addition, we apply the model to the task of graded lexical entailment (Vulić et al., 2016), and we assess the capability of *HyperVec* on generalizing hypernymy by mapping to German and Italian. Results on benchmark datasets of hypernymy show that the hierarchical embeddings outperform state-of-the-art measures and previous embedding models. Furthermore, the implementation of our models is made publicly available.[1]

2 Related Work

Unsupervised hypernymy measures: A variety of directional measures for unsupervised hypernymy detection (Weeds and Weir, 2003; Weeds et al., 2004; Clarke, 2009; Kotlerman et al., 2010; Lenci and Benotto, 2012) all rely on some variation of the *distributional inclusion hypothesis*: If u is a semantically narrower term than v, then a significant number of salient distributional features of u is expected to be included in the feature vector of v as well. In addition, Santus et al. (2014) proposed the *distributional informativeness hypothesis*, that hypernyms tend to be less informative than hyponyms, and that they occur in more general contexts than their hyponyms. All of these approaches represent words as vectors in distributional semantic models (Turney and Pantel, 2010), relying on the *distributional hypothesis* (Harris, 1954; Firth, 1957). For evaluation, these directional models use the AP measure to assess the proportion of hypernyms at the top of a score-sorted list. In a different vein, Kiela et al. (2015) introduced three unsupervised methods drawn from visual properties of images to determine a concept's generality in hypernymy tasks.

Supervised hypernymy methods: The studies in this area are based on word embeddings which represent words as low-dimensional and real-valued vectors (Mikolov et al., 2013b; Pennington et al., 2014). Each hypernymy pair is encoded by some combination of the two word vectors, such as concatenation (Baroni et al., 2012) or difference (Roller et al., 2014; Weeds et al., 2014). Hypernymy is distinguished from other relations by using a classification approach, such as SVM or LR. Because word embeddings are trained for similar and symmetric vectors, it is however unclear whether the supervised methods do actually learn the asymmetry in hypernymy (Levy et al., 2015).

Hypernymy-specific embeddings: These approaches are closest to our work. Yu et al. (2015) proposed a dynamic distance-margin model to learn term embeddings that capture properties of hypernymy. The neural model is trained on the taxonomic relation data which is pre-extracted. The resulting term embeddings are fed to an SVM classifier to predict hypernymy. However, this model only learns term pairs without considering their contexts, leading to a lack of generalization for term embeddings. Tuan et al. (2016) introduced a dynamic weighting neural network to learn term embeddings that encode information about hypernymy and also about their contexts, considering all words between a hypernym and its

[1] www.ims.uni-stuttgart.de/data/hypervec

hyponym in a sentence. The proposed model is trained on a set of hypernym relations extracted from WordNet (Miller, 1995). The embeddings are applied as features to detect hypernymy, using an SVM classifier. Tuan et al. (2016) handles the drawback of the approach by Yu et al. (2015), considering the contextual information between two terms; however the method still is not able to determine the directionality of a hypernym pair. Vendrov et al. (2016) proposed a method to encode order into learned distributed representations, to explicitly model partial order structure of the visual-semantic hierarchy or the hierarchy of hypernymy in WordNet. The resulting vectors are used to predict the transitive hypernym relations in WordNet.

3 Hierarchical Embeddings

In this section, we present our model of hierarchical embeddings *HyperVec*. Section 3.1 describes how we learn the embeddings for hypernymy, and Section 3.2 introduces the unsupervised measure *HyperScore* that is applied to the hypernymy tasks.

3.1 Learning Hierarchical Embeddings

Our approach makes use of a set of hypernyms which could be obtained from either exploiting the transitivity of the hypernymy relation (Fallucchi and Zanzotto, 2011) or lexical databases, to learn hierarchical embeddings. We rely on WordNet, a large lexical database of English (Fellbaum, 1998), and extract all hypernym–hyponym pairs for nouns and for verbs, including both direct and indirect hypernymy, e.g., *animal–bird, bird–robin, animal–robin*. Before training our model, we exclude all hypernym pairs which appear in any datasets used for evaluation.

In the following, Section 3.1.1 first describes the Skip-gram model which is integrated into our model for optimization. Section 3.1.2 then describes the objective functions to train the hierarchical embeddings for hypernymy.

3.1.1 Skip-gram Model

The Skip-gram model is a word embeddings method suggested by Mikolov et al. (2013b). Levy and Goldberg (2014) introduced a variant of the Skip-gram model with negative sampling (SGNS), in which the objective function is defined as follows:

$$\mathrm{J}_{SGNS} = \sum_{w \in V_W} \sum_{c \in V_C} \mathrm{J}_{(w,c)} \quad (1)$$

$$\mathrm{J}_{(w,c)} = \#(w,c) \log \sigma(\vec{w}, \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D}[\log \sigma(-\vec{w}, \vec{c}_N)] \quad (2)$$

where the skip-gram with negative sampling is trained on a corpus of words $w \in V_W$ and their contexts $c \in V_C$, with V_W and V_C the word and context vocabularies, respectively. The collection of observed words and context pairs is denoted as D; the term $\#(w,c)$ refers to the number of times the pair (w,c) appeared in D; the term $\sigma(x)$ is the sigmoid function; the term k is the number of negative samples and the term c_N is the sampled context, drawn according to the empirical unigram distribution P.

3.1.2 Hierarchical Hypernymy Model

Vector representations for detecting hypernymy are usually encoded by standard first-order distributional co-occurrences. In this way, they are insufficient to differentiate hypernymy from other paradigmatic relations such as synonymy, meronymy, antonymy, etc. Incorporating directional measures of hypernymy to detect hypernymy by exploiting the common contexts of hypernym and hyponym improves this relation distinction, but still suffers from distinguishing between hypernymy and meronymy.

Our novel approach presents two solutions to deal with these challenges. First of all, the embeddings are learned in a specific order, such that the similarity score for hypernymy is higher than the similarity score for other relations. For example, the hypernym pair *animal–frog* will be assigned a higher cosine score than the co-hyponymy pair *eagle–frog*. Secondly, the embeddings are learned to capture the distributional hierarchy between hyponym and hypernym, as an indicator to differentiate between hypernym and hyponym. For example, given a hyponym–hypernym pair (p, q), we can exploit the Euclidean norms of $\vec{q}$ and $\vec{p}$ to differentiate between the two words, such that the Euclidean norm of the hypernym $\vec{q}$ is larger than the Euclidean norm of the hyponym $\vec{p}$.

Inspired by the distributional lexical contrast model in Nguyen et al. (2016) for distinguishing antonymy from synonymy, this paper proposes two objective functions to learn hierarchical embeddings for hypernymy. Before moving

to the details of the two objective functions, we first define the terms as follows: $\mathbb{W}(c)$ refers to the set of words co-occurring with the context c in a certain window-size; $\mathbb{H}(w)$ denotes the set of hypernyms for the word w; the two terms $\mathbb{H}^+(w,c)$ and $\mathbb{H}^-(w,c)$ are drawn from $\mathbb{H}(w)$, and are defined as follows:

$$\mathbb{H}^+(w,c) = \{u \in \mathbb{W}(c) \cap \mathbb{H}(w) : cos(\vec{w},\vec{c}) - cos(\vec{u},\vec{c}) \geq \theta\}$$
$$\mathbb{H}^-(w,c) = \{v \in \mathbb{W}(c) \cap \mathbb{H}(w) : cos(\vec{w},\vec{c}) - cos(\vec{v},\vec{c}) < \theta\}$$

where $cos(\vec{x},\vec{y})$ stands for the cosine similarity of the two vectors $\vec{x}$ and $\vec{y}$; θ is the margin. The set $\mathbb{H}^+(w,c)$ contains all hypernyms of the word w that share the context c and satisfy the constraint that the cosine similarity of pair (w,c) is higher than the cosine similarity of pair (u,c) within a max-margin framework θ. Similarly, the set $\mathbb{H}^-(w,c)$ represents all hypernyms of the word w with respect to the common context c in which the cosine similarity difference between the pair (w,c) and the pair (v,c) is within a min-margin framework θ. The two objective functions are defined as follows:

$$\mathrm{L}_{(w,c)} = \frac{1}{\#(w,u)} \sum\nolimits_{u \in \mathbb{H}^+(w,c)} \partial(\vec{w},\vec{u}) \quad (3)$$

$$\mathrm{L}_{(v,w,c)} = \sum\nolimits_{v \in \mathbb{H}^-(w,c)} \partial(\vec{v},\vec{w}) \quad (4)$$

where the term $\partial(\vec{x},\vec{y})$ stands for the cosine derivative of $(\vec{x},\vec{y})$; and ∂ then is optimized by the negative sampling procedure.

The objective function in Equation 3 minimizes the distributional difference between the hyponym w and the hypernym u by exploiting the common context c. More specifically, if the common context c is the distinctive characteristic of the hyponym w (i.e. the common context c is closer to the hyponym w than to the hypernym u), the objective function $\mathrm{L}_{(w,c)}$ tries to decrease the distributional generality of hypernym u by moving w closer to u. For example, given a hypernym-hyponym pair *animal–bird*, the context *flap* is a distinctive characteristic of *bird*, because almost every *bird* can flap, but not every *animal* can flap. Therefore, the context *flap* is closer to the hyponym *bird* than to the hypernym *animal*. The model then tries to move *bird* closer to *animal* in order to enforce the similarity between *bird* and *animal*, and to decrease the distributional generality of *animal*.

In contrast to Equation 3, the objective function in Equation 4 minimizes the distributional difference between the hyponym w and the hypernym v by exploiting the common context c, which is a distinctive characteristic of the hypernym v. In this case, the objective function $\mathrm{L}_{(v,w,c)}$ tries to reduce the distributional generality of hyponym w by moving v closer to w. For example, the context word *rights*, a distinctive characteristic of the hypernym *animal*, should be closer to *animal* than to *bird*. Hence, the model tries to move the hypernym *animal* closer to the hyponym *bird*. Given that hypernymy is an asymmetric and also a hierarchical relation, where each hypernym may contain several hyponyms, our objective functions updates simultaneously both the hypernym and all of its hyponyms; therefore, our objective functions are able to capture the hierarchical relations between the hypernym and its hyponyms. Moreover, in our model, the margin framework θ plays a role in learning the hierarchy of hypernymy, and in preventing the model from minimizing the distance of synonymy or antonymy, because synonymy and antonymy share many contexts.

In the final step, the objective function which is used to learn the hierarchical embeddings for hypernymy combines Equations 1, 2, 3, and 4 by the objective function in Equations 5 and 6:

$$\mathrm{J}_{(w,v,c)} = \mathrm{J}_{(w,c)} + \mathrm{L}_{(w,c)} + \mathrm{L}_{(v,w,c)} \quad (5)$$

$$\mathrm{J} = \sum_{w \in V_W} \sum_{c \in V_C} \mathrm{J}_{(w,v,c)} \quad (6)$$

3.2 Unsupervised Hypernymy Measure

HyperVec is expected to show the two following properties: (i) the hyponym and the hypernym are close to each other, and (ii) there exists a distributional hierarchy between hypernyms and their hyponyms. Given a hypernymy pair (u,v) in which u is the hyponym and v is the hypernym, we propose a measure to detect hypernymy and to determine the directionality of hypernymy by using the hierarchical embeddings as follows:

$$HyperScore(u,v) = cos(\vec{u},\vec{v}) * \frac{\|\vec{v}\|}{\|\vec{u}\|} \quad (7)$$

where $cos(\vec{u},\vec{v})$ is the cosine similarity between $\vec{u}$ and $\vec{v}$, and $\|\cdot\|$ is the magnitude of the vector (or the Euclidean norm). The cosine similarity is applied to distinguish hypernymy from other re-

lations, due to the first property of the hierarchical embeddings, while the second property is used to decide about the directionality of hypernymy, assuming that the magnitude of the hypernym is larger than the magnitude of the hyponym. Note that the proposed hypernymy measure is unsupervised when the resource is only used to learn hierarchical embeddings.

4 Experiments

In this section, we first describe the experimental settings in our experiments (Section 4.1). We then evaluate the performance of *HyperVec* on three different tasks: i) unsupervised hypernymy detection and directionality (Section 4.2), where we assess *HyperVec* on ranking and classifying hypernymy; ii) supervised hypernymy detection (Section 4.3), where we apply supervised classification to detect hypernymy; iii) graded lexical entailment (Section 4.4), where we predict the strength of hypernymy pairs.

4.1 Experimental Settings

We use the ENCOW14A corpus (Schäfer and Bildhauer, 2012; Schäfer, 2015) with approx. 14.5 billion tokens for training the hierarchical embeddings and the default SGNS model. We train our model with 100 dimensions, a window size of 5, 15 negative samples, and 0.025 as the learning rate. The threshold θ is set to 0.05. The hypernymy resource for nouns comprises $105,020$ hyponyms, $24,925$ hypernyms, and $1,878,484$ hyponym–hypernym pairs. The hypernymy resource for verbs consists of $11,328$ hyponyms, $4,848$ hypernyms, and $130,350$ hyponym–hypernym pairs.

4.2 Unsupervised Hypernymy Detection and Directionality

In this section, we assess our model on two experimental setups: i) a ranking retrieval setup that expects hypernymy pairs to have a higher similarity score than instances from other semantic relations; ii) a classification setup that requires both hypernymy detection and directionality.

4.2.1 Ranking Retrieval

Shwartz et al. (2017) conducted an extensive evaluation of a large number of unsupervised distributional measures for hypernymy ranking retrieval proposed in previous work (Weeds and Weir, 2003; Santus et al., 2014; Clarke, 2009; Kotlerman et al., 2010; Lenci and Benotto, 2012; Santus et al., 2016). The evaluation was performed on four semantic relation datasets: **BLESS** (Baroni and Lenci, 2011), **WEEDS** (Weeds et al., 2004), **EVALUTION** (Santus et al., 2015), and **LENCI&BENOTTO** (Benotto, 2015). Table 1 describes the detail of these datasets in terms of the semantic relations and the number of instances. The Average Precision (AP) ranking measure is used to evaluate the performance of the measures.

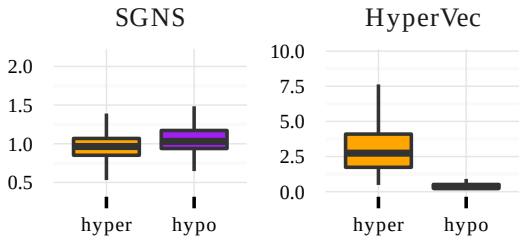
Dataset	Relation	#Instance	Total
BLESS	hypernymy	1,337	26,554
	meronymy	2,943	
	coordination	3,565	
	event	3,824	
	attribute	2,731	
	random-n	6,702	
	random-j	2,187	
	random-v	3,265	
EVALution	hypernymy	3,637	13,465
	meronymy	1,819	
	attribute	2,965	
	synonymy	1,888	
	antonymy	3,156	
Lenci&Benotto	hypernymy	1,933	5,010
	synonymy	1,311	
	antonymy	1,766	
Weeds	hypernymy	1,469	2,928
	coordination	1,459	

Table 1: Details of the semantic relations and the number of instances in each dataset.

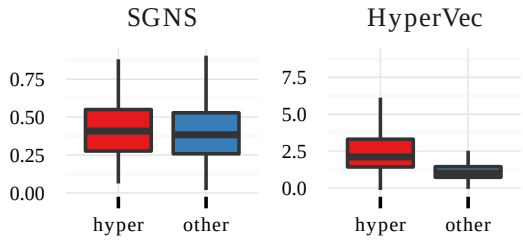
Dataset	Hypernymy vs.	Baseline	HyperScore
EVALution	other relations	0.353	**0.538**
	meronymy	0.675	**0.811**
	attribute	0.651	**0.800**
	antonymy	0.55	**0.743**
	synonymy	0.657	**0.793**
BLESS	other relations	0.051	**0.454**
	meronymy	0.76	**0.913**
	coordination	0.537	**0.888**
	attribute	0.74	**0.918**
	event	**0.779**	0.620
Lenci&Benotto	other relations	0.382	**0.574**
	antonymy	0.624	**0.696**
	synonymy	0.725	**0.751**
Weeds	coordination	0.441	**0.850**

Table 2: AP results of *HyperScore* in comparison to state-of-the-art measures.

In comparison to the state-of-the-art unsupervised measures compared by Shwartz et al. (2017) (henceforth, baseline models), we apply our unsupervised measure *HyperScore* (Equation 7) to rank hypernymy against other relations. Table 2



(a) Directionality task: hypernym vs. hyponym.



(b) Hypernymy detection: hypernymy vs. other relations.

Figure 1: Comparing *SGNS* and *HyperVec* on binary classification tasks. The y-axis shows the magnitude values of the vectors.

presents the results of using *HyperScore* vs. the best baseline models, across datasets. When detecting hypernymy among all other relations (which is the most challenging task), *HyperScore* significantly outperforms all baseline variants on all datasets. The strongest difference is reached on the BLESS dataset, where *HyperScore* achieves an improvement of 40% AP score over the best baseline model. When ranking hypernymy in comparison to a single other relation, *HyperScore* also improves over the baseline models, except for the *event* relation in the BLESS dataset. We assume that this is due to the different parts-of-speech (adjective and noun) involved in the relation, where *HyperVec* fails to establish a hierarchy.

4.2.2 Classification

In this setup, we rely on three datasets of semantic relations, which were all used in various state-of-the-art approaches before, and brought together for hypernymy evaluation by Kiela et al. (2015). **(i)** A subset of **BLESS** contains 1,337 hyponym-hypernym pairs. The task is to predict the directionality of hypernymy within a binary classification. Our approach requires no threshold; we only need to compare the magnitudes of the two words and to assign the hypernym label to the word with the larger magnitude. Figure 1a indicates that the magnitude values of the *SGNS* model cannot distinguish between a hyponym and a hypernym, while the hierarchical embeddings provide a larger magnitude for the hypernym. **(ii)** Following Weeds et al. (2014), we conduct a binary classification with a subset of 1,168 BLESS word pairs. In this dataset (**WBLESS**), one class is represented by hyponym–hypernym pairs, and the other class is a combination of reversed hypernym–hyponym pairs, plus additional holonym-meronym pairs, co-hyponyms and randomly matched nouns. For this classification we make use of our *HyperScore* measure that ranks hypernymy pairs higher than other relation pairs. A threshold decides about the splitting point between the two classes: *hyper* vs. *other*. Instead of using a manually defined threshold as done by Kiela et al. (2015), we decided to run 1 000 iterations which randomly sampled only 2% of the available pairs for learning a threshold, using the remaining 98% for test purposes. We present average accuracy results across all iterations. Figure 1b compares the default cosine similarities between the relation pairs (as applied by *SGNS*) and *HyperScore* (as applied by *HyperVec*) on this task. Using *HyperScore*, the class "hyper" can clearly be distinguished from the class "other". **(iii) BIBLESS** represents the most challenging dataset; the relation pairs from WBLESS are split into three classes instead of two: hypernymy pairs, reversed hypernymy pairs, and other relation pairs. In this case, we perform a three-way classification. We apply the same technique as used for the WBLESS classification, but in cases where we classify *hyper* we additionally classify the hypernymy direction, to decide between hyponym–hypernym pairs and reversed hypernym–hyponym pairs.

	BLESS	WBLESS	BIBLESS
Kiela et al. (2015)	0.88	0.75	0.57
Santus et al. (2014)	0.87	—	—
Weeds et al. (2014)	—	0.75	—
SGNS	0.44	0.48	0.34
HyperVec	**0.92**	**0.87**	**0.81**

Table 3: Accuracy for hypernymy directionality.

Table 3 compares our results against related

work. *HyperVec* outperforms all other methods on all three tasks. In addition we see again that an unmodified *SGNS* model cannot solve any of the three tasks.

4.3 Supervised Hypernymy Detection

For supervised hypernymy detection, we make use of the two datasets: the full **BLESS** dataset, and **ENTAILMENT** (Baroni et al., 2012), containing 2,770 relation pairs in total, including 1,385 hypernym pairs and 1,385 other relations pairs. We follow the same procedure as Yu et al. (2015) and Tuan et al. (2016) to assess *HyperVec* on the two datasets. Regarding BLESS, we extract pairs for four types of relations: hypernymy, meronymy, co-hyponymy (or *coordination*), and add the random relation for nouns. For the evaluation, we randomly select one concept and its relatum for testing, and train the supervised model on the 199 remaining concepts and its relatum. We then report the average accuracy across all concepts. For the ENTAILMENT dataset, we randomly select one hypernym pair for testing and train on all remaining hypernym pairs. Again, we report the average accuracy across all hypernyms.

We apply an SVM classifier to detect hypernymy based on *HyperVec*. Given a hyponym–hypernym pair (u, v), we concatenate four components to construct the vector for a pair (u, v) as follows: the vector difference between hypernym and hyponym ($\vec{v}-\vec{u}$); the cosine similarity between the hypernym and hyponym vectors ($cos(\vec{u}, \vec{v})$); the magnitude of the hyponym ($\|\vec{u}\|$); and the magnitude of the hypernym ($\|\vec{v}\|$). The resulting vector is fed into the SVM classifier to detect hypernymy. Similar to the two previous works, we train the SVM classifier with the RBF kernel, $\lambda = 0.03125$, and the penalty $C = 8.0$.

Table 4 shows the performance of *HyperVec* and the two baseline models reported by Tuan et al. (2016). *HyperVec* slightly outperforms the method of Tuan et al. (2016) on the BLESS dataset, and is equivalent to the performance of their method on the ENTAILMENT dataset. In comparison to the method of Yu et al. (2015), *HyperVec* achieves significant improvements.

4.4 Graded Lexical Entailment

In this experiment, we apply *HyperVec* to the dataset of graded lexical entailment, *HyperLex*, as introduced by Vulić et al. (2016). The *HyperLex* dataset provides soft lexical entailment on a continuous scale, rather than simplifying into a binary decision. *HyperLex* contains 2,616 word pairs across seven semantic relations and two word classes (nouns and verbs). Each word pair is rated by a score that indicates the strength of the semantic relation between the two words. For example, the score of the hypernym pair *duck–animal* is 5.9 out of 6.0, while the score of the reversed pair *animal–duck* is only 1.0.

Models	BLESS	ENTAILMENT
Yu et al. (2015)	0.90	0.87
Tuan et al. (2016)	0.93	0.91
HyperVec	**0.94**	0.91

Table 4: Classification results for BLESS and ENTAILMENT in terms of accuracy.

We compared *HyperScore* against the most prominent state-of-the-art hypernymy and lexical entailment models from previous work:

- Directional entailment measures (DEM) (Weeds and Weir, 2003; Weeds et al., 2004; Clarke, 2009; Kotlerman et al., 2010; Lenci and Benotto, 2012)
- Generality measures (SQLS) (Santus et al., 2014)
- Visual generality measures (VIS) (Kiela et al., 2015)
- Consideration of concept frequency ratio (FR) (Vulić et al., 2016)
- WordNet-based similarity measures (WN) (Wu and Palmer, 1994; Pedersen et al., 2004)
- Order embeddings (OrderEmb) (Vendrov et al., 2016)
- Skip-gram embeddings (SGNS) (Mikolov et al., 2013b; Levy and Goldberg, 2014)
- Embeddings fine-tuned to a paraphrase database with linguistic constraints (PARAGRAM) (Mrkšić et al., 2016)
- Gaussian embeddings (Word2Gauss) (Vilnis and McCallum, 2015)

The performance of the models is assessed through Spearman's rank-order correlation coefficient ρ (Siegel and Castellan, 1988), comparing the ranks of the models' scores and the human judgments for the given word pairs.

Measures		Embeddings	
Model	ρ	**Model**	ρ
FR	0.279	SGNS	0.205
DEM	0.180	PARAGRAM	0.320
SLQS	0.228	OrderEmb	0.191
WN	0.234	Word2Gauss	0.206
VIS	0.209	*HyperScore*	**0.540**

Table 5: Results (ρ) of *HyperScore* and state-of-the-art measures and word embedding models on graded lexical entailment.

Table 5 shows that *HyperScore* significantly outperforms both state-of-the-art measures and word embedding models. *HyperScore* outperforms even the previously best word embedding model PARAGRAM by .22, and the previously best measures FR by .27. The reason that *HyperVec* outperforms all other models is that the hierarchy between hypernym and hypornym within *HyperVec* differentiates hyponym–hypernym pairs from hypernym–hyponym pairs. For example, the *HyperScore* for the pairs *duck–animal* and *animal–duck* are 3.02 and 0.30, respectively. Thus, the magnitude proportion of the hypernym–hyponym pair *duck–animal* is larger than that for the pair *animal–duck*.

5 Generalizing Hypernymy

Having demonstrated the general abilities of *HyperVec*, this final section explores its potential for generalization in two different ways, (i) by relying on a small seed set only, rather than using a large set of training data; and (ii) by projecting *HyperVec* to other languages.

Hypernymy Seed Generalization: We utilize only a small hypernym set from the hypernymy resource to train *HyperVec*, relying on 200 concepts from the BLESS dataset. The motivation behind using these concepts is threefold: i) these concepts are distinct and unambiguous noun concepts; ii) the concepts were equally divided between living and non-living entities; iii) concepts have been grouped into 17 broader classes. Based on the seed set, we collected the hyponyms of each concept from WordNet, and then re-trained *HyperVec*. On the hypernymy ranking retrieval task (Section 4.2.1), *HyperScore* outperforms the baselines across all datasets (cf. Table 1) with AP values of 0.39, 0.448, and 0.585 for EVALution, LenciBenotto, and Weeds, respectively. For the graded lexical entailment task (Section 4.4), *HyperScore* obtains a correlation of $\rho = 0.30$, outperforming all models except for PARAGRAM with $\rho = 0.32$. Overall, the results show that *HyperVec* is indeed able to generalize hypernymy from small seeds of training data.

Generalizing Hypernymy across Languages: We assume that hypernymy detection can be improved across languages by projecting representations from any arbitrary language into our modified English *HyperVec* space. We conduct experiments for German and Italian, where the language-specific representations are obtained using the same hyper-parameter settings as for our English *SGNS* model (cf. Section 4.1). As corpus resource we relied on Wikipedia dumps[2]. Note that we do not use any additional resource, such as the German or Italian WordNet, to tune the embeddings for hypernymy detection. Based on the representations, a mapping function between a source language (German, Italian) and our English *HyperVec* space is learned, by relying on the least-squares error method from previous work using cross-lingual data (Mikolov et al., 2013a) and different modalities (Lazaridou et al., 2015).

To learn a mapping function between two languages, a one-to-one correspondence (word translations) between two sets of vectors is required. We obtained these translations by using the parallel Europarl[3] V7 corpus for German–English and Italian–English. Word alignment counts were extracted using *fast_align* (Dyer et al., 2013). We then assigned each source word to the English word with the maximum number of alignments in the parallel corpus. We could match 25,547 pairs for DE→EN and 47,475 pairs for IT→EN.

Taking the aligned subset of both spaces, we assume that X is the matrix obtained by concatenating all source vectors, and likewise Y is the matrix obtained by concatenating all corresponding English elements. Applying the $\ell 2$-regularized least-squares error objective can be described using the following equation:

$$\hat{\mathbf{W}} = \underset{\mathbf{W} \in \mathbb{R}^{d1 \times d2}}{\operatorname{argmin}} \|\mathbf{XW} - \mathbf{Y}\| + \lambda \|\mathbf{W}\| \quad (8)$$

Although we learn the mapping only on a subset of aligned words, it allows us to project every word in

[2]The Wikipedia dump for German and Italian were both downloaded in January 2017.

[3]http://www.statmt.org/europarl/

a source vocabulary to its English *HyperVec* position by using **W**.

Finally we compare the original representations and the mapped representation on the hypernymy ranking retrieval task (similar to Section 4.2.1). As gold resources we relied on German and Italian nouns pairs. For German we used the 282 German pairs collected via Amazon Mechanical Turk by Scheible and Schulte im Walde (2014). The 1,350 Italian pairs were collected via Crowdflower by Sucameli (2015) in the same way. Both collections contain hypernymy, antonymy and synonymy pairs. As before, we evaluate the ranking by AP, and we compare the cosine of the unmodified default representations against the *HyperScore* of the projected representations.

German	Hyp/All	Hyp/Syn	Hyp/Ant
DE-*SGNS*	0.28	0.48	0.40
DE→EN*HyperVec*	**0.37**	**0.65**	**0.47**
Italian			
IT-*SGNS*	0.38	0.50	0.60
IT→EN*HyperVec*	**0.44**	**0.57**	**0.65**

Table 6: AP results across languages, comparing *SGNS* and the projected representations.

The results are shown in Table 6. We clearly see that for both languages the default *SGNS* embeddings do not provide higher similarity scores for hypernymy pairs (except for Italian Hyp/Ant), but both languages provide higher scores when we map the embeddings into the English *HyperVec* space.

6 Conclusion

This paper proposed a novel neural model *HyperVec* to learn hierarchical embeddings for hypernymy. *HyperVec* has been shown to strengthen hypernymy similarity, and to capture the distributional hierarchy of hypernymy. Together with a newly proposed unsupervised measure *HyperScore* our experiments demonstrated (i) significant improvements against state-of-the-art measures, and (ii) the capability to generalize hypernymy and learn the relation instead of memorizing *prototypical hypernyms*.

Acknowledgments

The research was supported by the Ministry of Education and Training of the Socialist Republic of Vietnam (Scholarship 977/QD-BGDDT; Kim-Anh Nguyen), the DFG Collaborative Research Centre SFB 732 (Kim-Anh Nguyen, Maximilian Köper, Ngoc Thang Vu), and the DFG Heisenberg Fellowship SCHU-2580/1 (Sabine Schulte im Walde). We would like to thank three anonymous reviewers for their comments and suggestions.

References

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 (EACL)*, pages 23–32, Avignon, France.

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 (GEMS)*, pages 1–10, Edinburgh, Scotland.

Giulia Benotto. 2015. *Distributional models for semantic relations: A study on hyponymy and antonymy*. Ph.D. thesis, University of Pisa.

Or Biran and Kathleen McKeown. 2013. Classifying taxonomic relations between pairs of wikipedia articles. In *Proceddings of Sixth International Joint Conference on Natural Language Processing (IJCNLP)*, pages 788–794, Nagoya, Japan.

Daoud Clarke. 2009. Context-theoretic semantics for natural language: An overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics (GEMS)*, pages 112–119, Athens, Greece.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 644–648, Atlanta, USA.

Francesca Fallucchi and Fabio Massimo Zanzotto. 2011. Inductive probabilistic taxonomy learning using singular value decomposition. *Natural Language Engineering*, 17(1):71–94.

Christiane Fellbaum, editor. 1998. *WordNet – An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

John R. Firth. 1957. *Papers in Linguistics 1934-51*. Longmans, London, UK.

Maayan 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 (ACL)*, pages 107–114, Michigan, US.

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

Douwe Kiela, Laura Rimell, Ivan Vulić, and Stephen Clark. 2015. Exploiting image generality for lexical entailment detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL)*, pages 119–124, Beijing, China.

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

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and Pollution: Delving into Cross-Space Mapping for Zero-Shot Learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 270–280, Beijing, China.

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)*, pages 75–79, Montréal, Canada.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceddings of the 27th International Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 2177–2185, Montréal, Canada.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 970–976, Denver, Colorado.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting Similarities among Languages for Machine Translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119, Lake Tahoe, Nevada, US.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

George A. Miller and Christiane Fellbaum. 1991. Semantic networks of english. *Cognition*, 41:197–229.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, M. Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 142–148, San Diego, California.

Gregory Murphy. 2002. *The Big Book of Concepts*. MIT Press, Cambridge, MA, USA.

Roberto Navigli, Paola Velardi, and Stefano Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1872–1877, Barcelona, Catalonia, Spain.

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)*, pages 454–459, Berlin, Germany.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet: : Similarity - measuring the relatedness of concepts. In *Proceedings of the 19th National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence (AAAI)*, pages 1024–1025, California, USA.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Laura Rimell. 2014. Distributional lexical entailment by topic coherence. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 511–519, Gothenburg, Sweden.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 1025–1036, Dublin, Ireland.

Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016. Unsupervised measure of word similarity: How to outperform co-occurrence and vector cosine in vsms. In *Proceedings of the Thirtieth Conference on Artificial Intelligence AAAI)*, pages 4260–4261, Arizona, USA.

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 (EACL)*, pages 38–42, Gothenburg, Sweden.

Enrico Santus, Frances Yung, Alessandro Lenci, 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: Resources and Applications*, Beijing, China.

Roland Schäfer. 2015. Processing and querying large web corpora with the COW14 architecture. In *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora*, pages 28–34, Lancaster, UK.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 486–493, Istanbul, Turkey.

Silke Scheible and Sabine Schulte im Walde. 2014. A Database of Paradigmatic Semantic Relation Pairs for German Nouns, Verbs, and Adjectives. In *Proceedings of Workshop on Lexical and Grammatical Resources for Language Processing*, pages 111–119, Dublin, Ireland.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Valencia, Spain.

Sidney Siegel and N. John Castellan. 1988. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, Boston, MA.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 801–808, Sydney, Australia.

Irene Sucameli. 2015. Analisi computazionale delle relazioni semantiche: Uno studio della lingua italiana. B.s. thesis, University of Pisa.

Luu Anh Tuan, Yi Tay, Siu Cheung Hui, and See Kiong Ng. 2016. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 403–413, Austin, Texas.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-embeddings of images and language. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico.

Luke Vilnis and Andrew McCallum. 2015. Word representations via gaussian embedding. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, California, USA.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2016. Hyperlex: A large-scale evaluation of graded lexical entailment. *arXiv*.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David J. Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 2249–2259, Dublin, Ireland.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 81–88, Stroudsburg, PA, USA.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 1015–1021, Geneva, Switzerland.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 133–138, Las Cruces, New Mexico.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning term embeddings for hypernymy identification. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI)*, pages 1390–1397, Buenos Aires, Argentina.

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