Dual Tensor Model for Detecting Asymmetric Lexico-Semantic Relations

Goran Glavaš and **Simone Paolo Ponzetto**
Data and Web Science Group
University of Mannheim
B6, 26, DE-68159 Mannheim, Germany
{goran, simone}@informatik.uni-mannheim.de

Abstract

Detection of lexico-semantic relations is one of the central tasks of computational semantics. Although some fundamental relations (e.g., hypernymy) are asymmetric, most existing models account for asymmetry only implicitly and use the same concept representations to support detection of symmetric and asymmetric relations alike. In this work, we propose the Dual Tensor model, a neural architecture with which we explicitly model the asymmetry and capture the translation between unspecialized and specialized word embeddings via a pair of tensors. Although our Dual Tensor model needs only unspecialized embeddings as input, our experiments on hypernymy and meronymy detection suggest that it can outperform more complex and resource-intensive models. We further demonstrate that the model can account for polysemy and that it exhibits stable performance across languages.

1 Introduction

Detection of semantic relations that hold between words is the central task of lexical semantics, tightly coupled with obtaining representations that capture meaning of words (Mikolov et al., 2013; Wieting et al., 2015; Mrkšić et al., 2016, *inter alia*). As such, robust detection of lexico-semantic relations may benefit virtually any natural language processing application.

Because lexico-semantic knowledge bases (KBs) like WordNet (Fellbaum, 1998) are general and of limited coverage, numerous methods for detecting lexico-semantic relations rely on distributional word representations obtained from large corpora. Although distributional models have evolved over time, from count-based (Landauer et al., 1998) and generative (Blei et al., 2003) to prediction-based (Mikolov et al., 2013), the similarity between distributional vectors still indicates only the abstract semantic association and not a precise semantic relation (e.g., vectors of antonyms may be as similar as vectors of synonyms).

Consequently, a number of approaches have been proposed for specializing distributional spaces for specific lexico-semantic relations, either by (1) modifying the learning objective or regularization of the original embedding model by incorporating linguistic constraints (Yu and Dredze, 2014; Kiela et al., 2015) or (2) retroactively fitting the pre-trained unspecialized embeddings to linguistic constraints (Faruqui et al., 2015; Mrkšić et al., 2016). However, these methods specialize distributional vector spaces primarily for detecting the symmetric relation of *semantic similarity* (i.e., graded synonymy) and not for asymmetric lexico-semantic relations such as *hypernymy* and *meronymy*. On the other hand, models for embedding KBs (Bordes et al., 2013; Socher et al., 2013; Yang et al., 2015) uniformly model both symmetric and asymmetric relations. They learn a single vector representation (i.e., embedding) for each KB concept, assuming implicitly that the same concept representation is equally useful for predicting symmetric and asymmetric relations alike.

Relation-specific learning-based models have, to the largest extent, targeted hypernymy. Distributional models predict the hypernymy relations by combining raw distributional vectors of concepts in a pair (Baroni et al., 2012; Roller et al., 2014; Santus et al., 2014), whereas path-based models base predictions on lexico-syntactic paths from co-occurrence contexts obtained from a large corpus (Snow et al., 2004; Nakashole et al., 2012; Shwartz et al., 2016). Shwartz et al. (2016) combine the path-based and distributional models to

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1756–1766
Copenhagen, Denmark, September 7–11, 2017.

reach state-of-the-art performance in hypernymy detection. Both distributional and path-based methods, however, model asymmetry only implicitly (e.g., via the order of embeddings in the concatenation). Besides, path-based models are language-dependent since they require syntactically preprocessed data as input.

In this work, we propose the Dual Tensor model, a neural architecture that (1) models asymmetry more explicitly than existing models and (2) explicitly captures the translation of unspecialized distributional vectors into specialized embeddings better suited to detect the asymmetric relation of interest. The Dual Tensor model can be considered distributional as it requires only distributional vectors of words as input. Consequently, in contrast to path-based methods, it is language-independent and more widely applicable. Experimental results on hypernymy and meronymy detection show that the Dual Tensor model outperforms both distributional and path-based models. We additionally demonstrate that our approach exhibits stable performance across languages and can, to some extent, diminish the negative effects of polysemy.

2 Related Work

Specializing Word Embeddings. Unspecialized word embeddings (Mikolov et al., 2013; Pennington et al., 2014) capture general semantic properties of words, but are unable to differentiate between different types of semantic relations (e.g., vectors of *car* and *driver* might be as similar as vectors of *car* and *vehicle*). However, we often need embeddings to be similar only if an exact lexico-semantic relation holds between the words. Numerous methods for specializing word embeddings for particular relations have been proposed (Yu and Dredze, 2014; Faruqui et al., 2015; Kiela et al., 2015; Mrkšić et al., 2016, *inter alia*), primarily aiming to differentiate synonymic similarity from other types of semantic relatedness.

Some methods modify the objective or regularization of general embedding algorithms like CBOW or skip-gram (Mikolov et al., 2013) in order to directly train relation-specific embeddings from large corpora. Yu and Dredze (2014) extend the CBOW objective with synonymy constraints from WordNet and Paraphrase Database (PPDB) (Ganitkevitch et al., 2013). Similarly, Kiela et al. (2015) add synonyms as additional contexts for the skip-gram objective.

Other models update the whole unspecialized embedding space by moving closer together vectors of words standing in a particular relation. Starting with unspecialized embeddings of concepts, Faruqui et al. (2015) run a belief propagation algorithm on a graph induced from WordNet or PPDB. Wieting et al. (2015) couple an objective maximizing the similarity of PPDB pairs with the smart selection of the negative examples. Mrkšić et al. (2016) take this idea further by using antonym pairs from WordNet as negative examples.

All aforementioned models either directly train specialized embeddings or derive them by updating the unspecialized embeddings. In contrast, via dual tensors, we explicitly capture the function that transforms unspecialized embeddings to specialized embeddings that are better suited to detect the asymmetric relation of interest.

Embedding Knowledge Graphs. Recently, various models for embedding KB concepts and relations have been proposed (Bordes et al., 2013; Socher et al., 2013; Yang et al., 2015; Nickel et al., 2016, *inter alia*). These models predict existence of relations between entities by arithmetically combining concept vectors and relation matrices or tensors. The scoring functions of KG embedding models combine the concept embeddings via linear product (i.e., relation tensor multiplies the concatenation of concept vectors of the two entities) (Bordes et al., 2011), bilinear product (i.e., relation tensor first multiplies the left concept embedding and the result multiplies the embedding of the second concept) (Yang et al., 2015), or the combination of the two (Socher et al., 2013). Both linear and bilinear scoring functions implicitly model asymmetry as they are not commutative with respect to concept embeddings. In this work, we choose to leverage the bilinear product in our model, following the findings of Yang et al. (2015) who report bilinear product outperforming other scoring combinations.

KG embedding models employ the same concept embeddings for predicting all relations, symmetric and asymmetric alike. By directly updating concept embeddings in training, they cannot make relation predictions for concepts outside of the training set.

Hypernymy and Meronymy Detection. Hypernymy and meronymy are arguably the two most prominent asymmetric lexico-semantic relations. Methods for their detection can roughly be classified as either distributional or path-based. Path-based methods consider lexico-syntactic paths connecting pairs of words in their co-occurrence contexts in large corpus. Early approaches, e.g., Hearst (1992) for hypernymy and Berland and Charniak (1999) for meronymy, exploited a small set of manually created lexico-syntactic patterns that imply a relation of interest (e.g., *a such as b*). Subsequent approaches looked at ways to eliminate the need for manual compilation of extraction patterns. Pantel and Pennacchiotti (2006) and Girju et al. (2006) proposed bootstrapping approaches to meronymy detection, starting from a seed set of part-whole pairs. Snow et al. (2004) provided all dependency paths connecting the concepts in corpus to a logistic regression classifier for hypernymy detection.

Distributional methods detect asymmetric relations using only distributional vectors of words as input. Distributional models come in both unsupervised and supervised flavors. Unsupervised metrics for hypernymy detection assume either that the hyponym's contexts are included in the hypernym's contexts (Weeds and Weir, 2003; Kotlerman et al., 2010) or that the linguistics contexts of a hyponym are more informative than the contexts of its hypernyms (Rimell, 2014; Santus et al., 2014). Supervised hypernymy classifiers represent the pair of words by combining their distributional vectors in different ways – concatenating them (Baroni et al., 2012) or subtracting them (Roller et al., 2014) – and feeding the resulting vector to a supervised classifier like logistic regression. Most recently, Shwartz et al. (2016) coupled path-based and distributional information with a recurrent neural network (RNN), yielding state-of-the-art hypernymy detection performance. Although our Dual Tensor model is purely distributional, we show that it may outperform such a hybrid model which additionally exploits syntactic information.

Distributional and path-based models have been used to discriminate between multiple lexico-semantic relations, including hypernymy and meronymy, at once (Santus et al., 2016; Shwartz and Dagan, 2016). However, as pointed out by (Chersoni et al., 2016), distributional vectors and scores based on their comparison fail to discriminate between multiple relation types at once. In this work, we focus on binary classification for a single relation (hypernymy and meronymy) at a time.

3 Dual Tensor Model

The following assumptions and desirable properties guided the design of the Dual Tensor model for detection of asymmetric lexico-semantic relations:

(1) Unspecialized distributional vectors are not good signals for detecting specific lexico-semantic relations. We thus need to derive specialized representations that are better suited for detecting the specific asymmetric relation of interest.

(2) The transformation from unspecialized distributional vectors of words to their relation-specialized embeddings should be captured explicitly, via a well-defined transformation function. Having an explicit embedding specialization function alleviates the need to specialize the entire unspecialized embedding space at once, like existing models do (Faruqui et al., 2015; Mrkšić et al., 2016).

(3) Each concept should have two different relation-specialized embeddings – one for each end of an asymmetric relation. For instance, for hypernymy, the concept's specialized embedding for pairs in which it is considered to be a hyponym (e.g., *dog* in *dog–animal*) should differ from its embedding in pairs in which it is tested as a hypernym (e.g., *dog* in *maltese–dog*).

(4) An unspecialized distributional vector of the word might – for each end of the asymmetric relation – be transformed into several specialized vectors instead of only one. This way the model may implicitly account for polysemy – i.e., different specialized vectors might capture asymmetric properties of different senses of polysemous words. E.g., the hyponym properties of *bank* in the pair *bank* vs. *building* may be different from those in the pair *bank* vs. *company*).

Figure 1 depicts the overall architecture of the Dual Tensor model, incorporating all four of above-mentioned design guidelines.

3.1 Dual Tensors

For a given pair of concepts (c_1, c_2), Dual Tensor model computes the score $s(c_1, c_2)$ indicating the likelihood that an asymmetric lexico-semantic relation holds between the concepts (e.g., for meronymy, how likely it is that c_1 is a *part of* c_2). The model takes as input the unspecialized embeddings of the two concepts, e_1 and e_2. For single-word concepts these are simply pre-trained word embeddings, whereas for multi-word concepts, similar to (Socher et al., 2013), we average the pre-trained embeddings of constituent words.

The unspecialized input embeddings are next translated into specialized embeddings, meant to

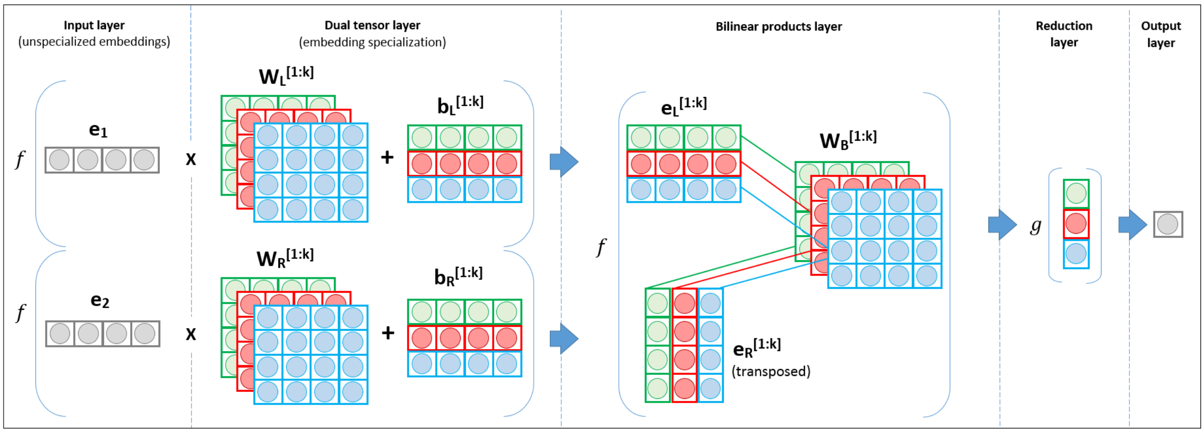

Figure 1: The architecture of the Dual Tensor model.

better capture the existence of the asymmetric relation between the concepts, via *specialization tensors*. By introducing dedicated tensors we – unlike existing models, which directly propagate updates to unspecialized embeddings (Faruqui et al., 2015; Mrkšić et al., 2016) – explicitly learn the specialization function. With an explicit specialization function, we do not have to specialize the whole embedding space at once. Also, unlike KG completion models (Bordes et al., 2013; Socher et al., 2013), we can make predictions for pairs involving concepts unseen in the training data.

We explicitly model asymmetry by introducing two specialization tensors (hence the model name) that differently specialize the unspecialized input embeddings of concepts. The left tensor, $\mathbf{W_L^{[1:k]}}$ (with the corresponding set of bias vectors $\mathbf{b_L^{[1:k]}}$), specializes the concept embedding if the concept is the first element of the pair, whereas the right tensor, $\mathbf{W_R^{[1:k]}}$ (with bias vectors $\mathbf{b_R^{[1:k]}}$), specializes the concept embedding when the concept is the second element of the pair:

$$\mathbf{e_L^{[1:k]}} = tanh\left(e_1 \mathbf{W_L^{[1:k]}} + \mathbf{b_L^{[1:k]}}\right)$$
$$\mathbf{e_R^{[1:k]}} = tanh\left(e_2 \mathbf{W_R^{[1:k]}} + \mathbf{b_R^{[1:k]}}\right)$$

When predicting hypernymy, for example, dual tensors ensure that the specialized representation for concept *cat* in pairs like *cat–animal* differs from its specialized representation in pairs like *birman–cat*.

Specialization tensors map an unspecialized embedding into a set of k specialized embeddings – each slice of the tensor, W_L^i (W_R^i), together with the corresponding bias vector b_L^i (b_R^i), produces one specialized vector e_L^i (e_R^i). By using specialization tensors with k slices instead of specialization matrices we make the model more general. The tensor-based model trivially degrades to the matrix-based model by setting $k = 1$. We obtain the final specialized representation of a concept by non-linearly transforming (hyperbolic tangent) the product of an unspecialized input embedding and the specialization tensor.[1]

3.2 Bilinear Product and Scoring

Using dual tensors, we transform unspecialized embeddings into asymmetrically specialized representations – sets of specialized vectors – which we next use to predict whether the asymmetric relation holds between the concepts. Our scoring function is based on bilinear products between (1) specialized vectors $\mathbf{e_L^{[1:k]}}$ of the first concept, (2) relation tensor $\mathbf{W_B^{[1:k]}}$, and (3) specialized vectors $\mathbf{e_R^{[1:k]}}$ of the second concept. For each pair of specialized vectors e_L^i and $e_R^i, i \in \{1, \ldots, k\}$, we compute the bilinear product score, using the corresponding slice W_B^i of the relation tensor $\mathbf{W_B^{[1:k]}}$:

$$b^i = e_L^i W_B^i (e_R^i)^T.$$

The final relation score $s(c_1, c_2)$ for a given pair of concepts is computed by reducing the vector of bilinear product scores $\mathbf{b}$ to the mean value (function g in Figure 1)[2] and non-linearly bounding the resulting score to the $[-1, 1]$ range:

$$s(c_1, c_2) = tanh\left(\frac{1}{k}\sum_{i=1}^{k} b^i\right).$$

[1] Preliminary experiments without applying a non-linear transformation yielded consistently poorer performance.

[2] We also experimented with min- and max-reduction, but the reduction to the mean yielded best preliminary results.

3.3 Optimization

Dual Tensor model is parametrized by the specialization tensors, their corresponding bias vectors, and the relation tensor, namely, $\mathbf{\Omega} = \{\mathbf{W}_\mathbf{L}^{[1:k]}, \mathbf{W}_\mathbf{R}^{[1:k]}, \mathbf{b}_\mathbf{L}^{[1:k]}, \mathbf{b}_\mathbf{R}^{[1:k]}, \mathbf{W}_\mathbf{B}^{[1:k]}\}$. Let A be the set of concept pairs in the training set, $A = \{p^i = (c_1^i, c_2^i)\}_{i=1}^N$. We learn model's parameters by minimizing the margin-based objective:

$$J(\mathbf{\Omega}) = \lambda \|\mathbf{\Omega}\|_2 + \sum_{p^i \in A} max\left(0, 1 - s(p^i) \cdot y(p^i)\right)$$

where $s(p^i)$ is model's prediction for the pair (c_1^i, c_2^i), $y(p^i) \in \{-1, 1\}$ is the true label of that pair, and λ is the regularization coefficient. In all our experiments, we trained the model in mini-batches, optimizing the parameters with the RMSProp algorithm (Tieleman and Hinton, 2012).

The model has three hyperparameters: the length of the unspecialized input embeddings l, the number of tensor slices k, and the regularization factor λ. We optimize the hyperparameters (together with the starting learning rate value) via grid-search, by maximizing performance on the validation portion of each dataset. In all our experiments, except the multilingual comparison (Section 5.3), we evaluated variants of the Dual Tensor model using pre-trained English GloVe word embeddings (Pennington et al., 2014) with varying length, $l \in \{50, 100, 200, 300\}$ and tensors with $k \in \{1, \ldots, 5\}$ slices. In most experiments, the optimal configuration was $l = 300$ and $k = 3$.

4 Evaluation

We evaluate the Dual Tensor model on several datasets for detecting hypernymy and meronymy, two arguably most prominent asymmetric lexico-semantic relations. In all experiments, we compare the model's performance with state-of-the-art results on respective datasets. Additionally, aiming to quantify the effects that different components of the Dual Tensor model have on prediction performance, we evaluate two reduced models variants.

4.1 Datasets

We evaluate the Dual Tensor model on the following hypernymy and meronymy detection datasets:

HypeNet dataset. Arguing that existing datasets were too small for training their recurrent network, Shwartz et al. (2016) compiled this dataset for hypernymy detection from several external KBs, taking only pairs of concepts in direct relation (i.e., no transitive closure).

Other hypernymy detection datasets. We additionally evaluate the Dual Tensor model on four smaller datasets for hypernymy detection: (1) BLESS dataset (Baroni and Lenci, 2011) and EVALuation dataset (Santus et al., 2015) contain instances of hypernymy and four other relations. BLESS additionally contains random word pairs; (2) Weeds dataset (Weeds et al., 2014) contains hypernymy and co-hyponymy pairs; (3) Benotto dataset (Benotto, 2015) couples hypernymy pairs with synonymy and antonymy pairs. Because these datasets contain at most several thousand pairs, we only use them to evaluate the performance of models trained on larger datasets;

WN-Hy and WN-Me datasets. We create these datasets by taking concept pairs from WordNet. We take all instances from the transitive closure of hypernymy (all parts of speech) and meronymy (nouns) relations and couple them with all synonym and antonym relations (all parts of speech), as well as lexical entailment relations (verbs).

For the WN-Hy dataset we designate all hypernymy relations (i.e., both direct and indirect) as positive instances and their inverses (i.e., hyponymy relations) together with all other relations as negative instances. Finally, we balance the dataset by randomly sampling negative instances to match the number of positive instances. Analogously, we create the WN-Me dataset by taking meronymy relations as positive instances. We compile three different WN-Hy datasets: WN-Hy-EN using English WordNet (Fellbaum, 1998), WN-Hy-ES using Spanish WordNet (Gonzalez-Agirre et al., 2012), and WN-Hy-FR using French WordNet (Sagot and Fišer, 2008). To allow for fair comparison of model's performance across languages, we randomly sample two larger dataset (English and French) to match in size the smallest (Spanish).

Lexical and Random Splits. Levy et al. (2015) showed that supervised distributional models for classifying lexico-semantic relations suffer from overfitting in settings with significant lexical overlap between the training and test set. In such settings models tend to learn properties of individual words (e.g., that a word is a prototypical hypernym) instead of relations between words. The reported results on such datasets are thus overly optimistic estimates of models' true performance.

Dataset	Train	Val.	Test
HypeNet (rand)	49.5K (20%)	3.5K (19%)	17.7K (20%)
HypeNet (lex)	20.3K (20%)	1.4K (20%)	6.6K (20%)
BLESS	–	2.7K (5%)	23.9K (5%)
EVALuation	–	1.4K (24%)	12.3K (27%)
Weeds	–	293 (50%)	2.6K (50%)
Benotto	–	501 (41%)	4.5K (38%)
WN-Hy-EN	103K (50%)	15K (50%)	30K (50%)
WN-Hy-EN	103K (50%)	15K (50%)	30K (50%)
WN-Hy-FR	103K (50%)	15K (50%)	30K (50%)
WN-Me (rand)	13.9K (50%)	2K (50%)	4K (50%)
WN-Me (lex)	7.9K (50%)	208 (50%)	318 (50%)

Table 1: Datasets used in evaluation.

	Lex. split			Rand. split		
Model	P	R	F_1	P	R	F_1
HypeNet path-based	69.1	63.2	66.0	81.1	71.6	76.1
HypeNet hybrid	**80.9**	61.7	70.0	91.3	**89.0**	**90.1**
Concat-SVM	75.4	55.1	63.7	90.1	63.7	74.6
Bilin-Prod	53.1	53.3	53.2	74.0	79.4	76.6
Single-T	68.4	70.0	69.2	84.8	86.7	85.7
Dual-T	70.5	**78.5**	**74.3**	**93.3**	82.6	87.6

Table 2: Hypernymy classification performance.

To eliminate the effect of lexical memorization, Levy et al. (2015) propose dataset splits with no lexical overlap between the train and test portions. However, model's performance in a lexically-split setting is an overly pessimistic estimate of models' true performance – in a realistic scenario, the model will occasionally make predictions for pairs involving some of the concepts from the training set. Because the true model performance is likely between the performance on a randomly-split and performance on a lexically-split dataset, we report models' performance in both of these settings.

We show the sizes of all dataset variants used in our experiments in Table 1. We additionally report the proportion of positive instances (in brackets), as this percentage directly affects some evaluation metrics (precision, F_1-score, average precision).

4.2 Baselines

In addition to specific models yielding best performance on particular datasets, we compare the Dual Tensor model (Dual-T) with these baselines:

Supervised distributional baseline (Concat-SVM). We train SVM model with RBF kernel on concatenation of unspecialized concept embeddings (Baroni et al., 2012), following Levy et al. (2015), who report this model outperforming other types of embedding composition;

Bilinear product (Bilin-Prod). This model is the simple bilinear product between the unspecialized concept embeddings, parametrized only by the relation matrix W_B. That is, the prediction score for a pair of concepts is given as $s(c_1, c_2) = e_1 W_B e_2^T$. The bilinear model implicitly captures asymmetry by learning a non-symmetric relation matrix W_B. By comparing the performances of Bilin-Prod and Dual-T, we jointly quantify the effects of (1) explicit modeling of asymmetry and (2) relation-specific embedding specialization;

Single tensor model (Single-T). This is the reduction of the Dual Tensor model in which we use only one specialization tensor, i.e., $\mathbf{W}_{\mathbf{L}}^{[\mathbf{1:k}]} = \mathbf{W}_{\mathbf{R}}^{[\mathbf{1:k}]}$. In other words, Single-T model always specializes the unspecialized embedding of a concept the same way, regardless of the concept's position in a candidate pair. By comparing the performance of the Dual-T model with that of Single-T, we measure the effect of asymmetrically specializing unspecialized embeddings.

Same as for the Dual-T model, we optimize the hyperparameters of the baselines on the validation portions of the datasets used for evaluation.

4.3 Classification Experiments

Binary classification is the most straightforward evaluation setting for relation detection models. For a pair of concepts, we make the binary asymmetric relation prediction $r_a(c_1, c_2)$ simply by thresholding the model's prediction scores, i.e., $r_a(c_1, c_2) = I\{s(c_1, c_2) > 0\}$, where I is the indicator function.

Hypernymy classification. We first evaluate the Dual-T model and the baselines on the HypeNet dataset (Shwartz et al., 2016). We show the performance of the Dual-T model in Table 2, together with the path-based and hybrid (combination of path-based and distributional signal) variants of the the state-of-the-art RNN model of Shwartz et al. (2016). On the more challenging, lexically-split dataset Dual-T model significantly[3] outperforms the more complex hybrid HypeNet model (Shwartz et al., 2016), an RNN model coupling representations of syntactic paths from a large corpus with

[3] All performance differences were tested using the non-parametric stratified shuffling test (Yeh, 2000) with $\alpha = 0.05$.

	Lex. split			Rand. split		
Model	P	R	F_1	P	R	F_1
CONCAT-SVM	**78.6**	44.6	56.9	79.9	75.9	77.9
BILIN-PROD	73.3	50.0	59.4	81.0	79.8	80.5
SINGLE-T	77.7	55.5	64.8	85.7	82.6	84.1
DUAL-T	76.5	**61.1**	**67.9**	**87.7**	**85.3**	**86.5**

Table 3: Meronymy classification performance.

unspecialized concept embeddings. In both settings DUAL-T outperforms SINGLE-T which, in turn, outperforms BILIN-PROD. This empirically justifies both our explicit modeling of asymmetry and relation-specific embedding specialization.

Meronymy classification. We next evaluate the meronymy classification performance of the models on the WN-Me dataset. The results are shown in Table 3. Same as in the case of hypernymy classification, DUAL-T significantly outperforms all three baselines, with SINGLE-T outperforming BILIN-PROD. All distributional models we evaluate achieve poorer performance on meronymy than hypernymy detection, especially considering that WN-Me is a balanced dataset, whereas HypeNet is heavily skewed towards negative instances.

4.4 Ranking Experiments

Shwartz et al. (2017) propose ranking as an alternative evaluation setting for hypernymy detection. The goal is to rank positive relation pairs higher than negative ones. Our DUAL-T model (and associated baselines) rank the concept pairs in decreasing order of assigned relations scores $s(c_1, c_2)$. Following Shwartz et al. (2017), we report performance in terms of overall average precision (AP) and average precision at rank 100 (AP@100).

Hypernymy ranking. We evaluate the ranking performance on four small hypernymy test sets: BLESS, EVALuation, Benotto, and Weeds (cf. Table 1). As these datasets are not big enough to train neural models, we train all models on the HypeNet dataset. For each test set we eliminate the lexical overlap by removing from the HypeNet dataset pairs containing any concept from that test set.

Table 4 displays ranking performance for DUAL-T model, the supervised baselines, and the best-performing unsupervised hypernymy detection score (BEST-UNSUP, performance taken from (Shwartz et al., 2017)). Hypernymy ranking results depict the effectiveness of the DUAL-T model with respect to supervised baselines even more clearly than hypernymy classification results. All supervised models outperform the best unsupervised model in terms of AP, but only DUAL-T is consistently better when considering only 100 top-ranked pairs (AP@100). This adds to the conclusion that explicit modeling of asymmetry using dual tensors yields crucial performance boost.

Meronymy ranking. We measure the ranking performance for meronymy detection on the WN-Me dataset, reporting the results for both randomly- and lexically-split variants of the dataset in Table 5. Meronymy ranking results are in line with performance figures for hypernymy ranking. Again, DUAL-T consistently outperforms all three baselines. Absolute AP scores for meronymy are higher than those we report for hypernymy, but this is merely because WN-Me is a balanced dataset, whereas the hypernymy ranking test sets (with the exception of the Weeds dataset) are substantially skewed in favor of negative concept pairs.

5 Analysis

We perform additional analyses, providing further insights into DUAL-T model's performance. We analyze how model's performance depends on concept distance in WordNet and on number of concept senses. We also examine the stability of DUAL-T model's performance across different languages.

5.1 WordNet Distance

Unlike the HypeNet dataset (Shwartz et al., 2016), which contains only pairs of concepts that exist in a direct relation in some external knowledge base, our WN-Hy and WN-Me datasets (cf. Section 4.1) contain pairs of concepts of varying distance in WordNet, allowing for a more fine-grained analysis of the Dual Tensor model's performance.

We divide the test sets of WN-Hy-EN and WN-Me into five buckets according to the shortest path distance between concepts in WordNet.[4] We show hypernymy and meronymy prediction accuracies for all buckets in Figure 2. For hypernymy, we observe significantly lower accuracy for pairs of concepts appearing close in WordNet hierarchy. Close hyponym-hypernym pairs (e.g., *car–vehicle*) tend to occur in similar contexts and consequently have similar unspecialized embeddings. Such hypernymy instances are difficult to discern from syn-

[4] For any concept with multiple senses, we considered the WordNet synset of its dominant sense.

Dataset	BLESS		EVALuation		Benotto		Weeds	
Model	AP	AP@100	AP	AP@100	AP	AP@100	AP	AP@100
Best-unsup (Shwartz et al., 2017)	.051	.540	.353	.661	.382	.617	.441	.911
Concat-SVM	.097	.235	.321	.329	.523	.586	.644	.793
Bilin-Prod	.277	.627	.355	.457	.477	.678	.712	.948
Single-T	.463	.777	.433	.668	.501	.605	.771	.958
Dual-T	**.487**	**.823**	**.446**	**.866**	**.557**	**.847**	**.774**	**.985**

Table 4: Hypernymy detection, ranking results.

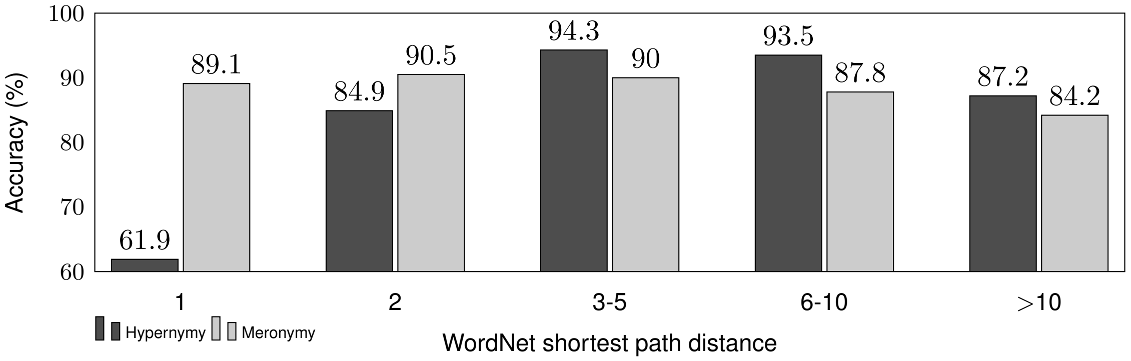

Figure 2: Hypernymy and meronymy performance with respect to WordNet shortest path distance.

	Lex. split		Rand. split	
Model	AP	AP@100	AP	AP@100
Concat-SVM	.686	.775	.796	.865
Bilin-Prod	.682	.832	.878	.947
Single-T	.772	.900	.909	.979
Dual-T	**.840**	**.967**	**.936**	**1.00**

Table 5: Meronymy detection, ranking results.

onymous pairs (e.g., *car–automobile*). The same effect is, however, not observed for meronymy – part-whole relations between close concepts are as detectable as between more distant concepts. This is probably because *part* concepts appear in different contexts than *whole* concepts (e.g., *wheel-car*), resulting in distinct unspecialized embeddings in the first place. For both relations we observe a drop in performance for pairs of very distant concepts. Such pairs typically contain one very abstract concept (e.g., *object*), but embeddings of abstract concepts are not superpositions of embeddings of their hyponyms (Rimell, 2014) nor their meronyms.

5.2 Effects of Polysemy

Given that our Dual Tensor model takes unspecialized concept embeddings as input and that unspecialized embeddings do not discern between different senses of words, our Dual Tensor model treats monosemous and polysemous concepts equally. Intuitively, predicting asymmetric relations for pairs involving polysemous concepts should be more difficult than for pairs of monosemous concepts, because the models in such cases additionally need to learn to discern between different concept senses.

While designing the Dual Tensor model, we hypothesized that different tensor slices might be able to accommodate for asymmetric relations involving different senses of polysemous words. In order to closer examine the effects of polysemy on the performance of the Dual Tensor model, we partitioned the test portions of the WN-Hy and WN-Me datasets according to number of senses of the concept pair (we average the number of senses of the two concepts in a candidate pair). We show the Dual Tensor model's performance ($k = 3, l = 300$) on different number-of-senses buckets, both for hypernymy and meronymy prediction, in Figure 3.

For hypernymy, the general trend is as expected: the larger the average number of senses of concepts in the candidate pair, the lower the prediction accuracy. The exception is the bucket $(3, 5]$ for which the performance is higher than for the previous bucket $(1, 3]$. The drop in performance is not drastic as long as the model is not dealing with highly

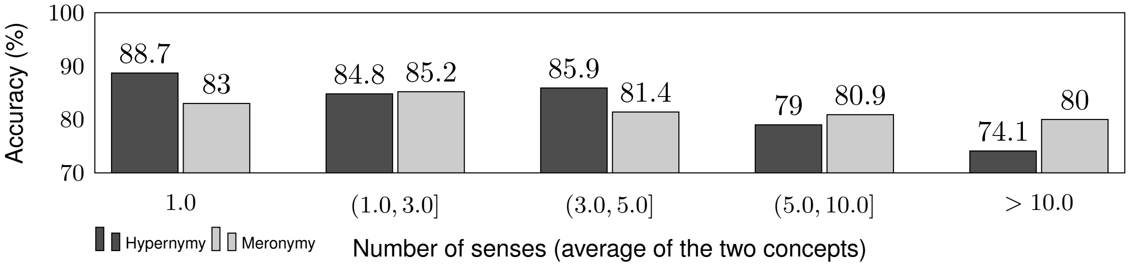

Figure 3: Hypernymy and meronymy performance with respect to concept polysemy.

polysemous concepts (with more than five senses). These performance figures suggest that, via the multiple tensor slices, the DUAL-T model can, to some extent, alleviate the effects that polysemy has on predicting asymmetric lexico-semantic relations.

Somewhat surprisingly, the polysemy seems not to have a clear negative effect for meronymy. Prediction accuracy on pairs of highly polysemous concepts seems to be similar to that on monosemous concept pairs. An instance-level inspection reveals that meronymy detection is more sensitive to the number of senses of the *part* candidate concept than of the *whole* concept. In other words, if we partition the test set only according to the number of senses of the *part* concept, then the trends are similar to those observed for hypernymy.

5.3 Multilingual Comparison

To examine how the Dual Tensor model performs across languages, we evaluate its performance on equally-sized hypernymy detection datasets in English, Spanish, and French (cf. Section 4.1 and Table 1). To increase the comparability of results, for each of the three languages we trained word embeddings using the CBOW algorithm (Mikolov et al., 2013) on the Wikipedia dump of respective language. Also, for all three models we select the hyperparameter configuration that turned out to be optimal most often in previous experiments – we set the length of unspecialized embeddings to $l = 300$ and number of tensor slices to $k = 3$. Hypernymy classification performance for different languages is shown in Table 6. The results suggest that Dual Tensor model exhibits stable performance across languages. The small performance differences between languages may be attributed to different sizes of respective Wikipedia dumps (on which we train unspecialized embeddings) as well as to inherent differences in language complexity (e.g., English being morpho-syntactically simpler).

Language	Dataset	P	R	F_1
English	WN-Hy-EN	89.9	86.1	87.9
Spanish	WN-Hy-ES	88.7	82.1	85.3
French	WN-Hy-FR	86.2	82.7	84.4

Table 6: Hypernymy classification performance for different languages.

6 Conclusion

We have presented a neural model for detecting asymmetric semantic relations. Unlike existing models, which uniformly treat asymmetric and symmetric relations, our Dual Tensor model captures asymmetry explicitly using a pair of specialization tensors that produce two different embedding specializations, depending on the concept's role in the relation. Instead of just updating unspecialized embeddings, with specialization tensors we also explicitly capture the mapping function.

The results from a battery of hypernymy and meronymy experiments show that via asymmetric specialization of concept embeddings the Dual Tensor model is able to outperform (1) the supervised model directly using unspecialized embeddings as well as (2) the more complex neural architecture that additionally exploits syntactic information. We have additionally shown that our model can diminish the negative effects of polysemy and that it exhibits stable performance across languages.

As future work, we plan to develop similar models based on explicit specialization tensors for detecting symmetric relations (e.g., synonymy, antonymy). We will also seek to exploit the Dual Tensor model in different downstream tasks, e.g., hypernymy detection for taxonomy induction (Faralli et al., 2017) or recognizing textual entailment.

Downloads. We make the code of the models and all datasets available at `https://bitbucket.org/gg42554/dual-tensors/`.

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*. pages 23–32.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the 2011 Workshop on GEometrical Models of Natural Language Semantics*. pages 1–10.

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

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.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3(Jan):993–1022.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 2013 Annual Conference on Neural Information Processing Systems*. pages 2787–2795.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*. pages 301–306.

Emmanuele Chersoni, Giulia Rambelli, and Enrico Santus. 2016. CogALex-V Shared Task: ROOT18. *CoRR* abs/1611.01101.

Stefano Faralli, Alexander Panchenko, Chris Biemann, and Simone Paolo Ponzetto. 2017. The ContrastMedium algorithm: Taxonomy induction from noisy knowledge graphs with just a few links. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. pages 590–600.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1606–1615.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 758–764.

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

Aitor Gonzalez-Agirre, Egoitz Laparra, and German Rigau. 2012. Multilingual central repository version 3.0: upgrading a very large lexical knowledge base. In *Proceedings of the 6th Global WordNet Conference*.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics: Volume 2*. pages 539–545.

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 2044–2048.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering* 16(04):359–389.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes* 25(2-3):259–284.

Omer Levy, Steffen Remus, Chris Biemann, Ido Dagan, and Israel Ramat-Gan. 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*. pages 970–976.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. pages 3111–3119.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. 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*. pages 142–148.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. Patty: a taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pages 1135–1145.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. pages 1955–1961.

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*. pages 113–120.

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*. pages 1532–1543.

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*. pages 511–519.

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*. pages 1025–1036.

Benoît Sagot and Darja Fišer. 2008. Building a free French WordNet from multilingual resources. In *Proceedings of the Ontolex 2008 Workshop*.

Enrico Santus, Anna Gladkova, Stefan Evert, and Alessandro Lenci. 2016. The CogALex-V shared task on the corpus-based identification of semantic relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon*. pages 69–79.

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: Volume 2, Short Papers*. pages 38–42.

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*. pages 64–69.

Vered Shwartz and Ido Dagan. 2016. Cogalex-V shared task: Lexnet-integrated path-based and distributional method for the identification of semantic relations. *arXiv preprint arXiv:1610.08694* .

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics: Volume 1, Long Papers*. pages 2389–2398.

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: Volume 1, Long Papers*. pages 65–75.

Rion Snow, Daniel Jurafsky, Andrew Y Ng, et al. 2004. Learning syntactic patterns for automatic hypernym discovery. In *Proceedings of the 2004 Annual Conference on Neural Information Processing Systems*. volume 17, pages 1297–1304.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 2013 Annual Conference on Neural Information Processing Systems*. pages 926–934.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4(2).

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

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of the 2003 conference on Empirical methods in Natural Language Processing*. pages 81–88.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics* 3:345–358.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 2015 International Conference on Learning Representations*.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th International Conference on Computational Linguistics*. pages 947–953.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. pages 545–550.