Understanding Feature Focus in Multitask Settings for Lexico-semantic Relation Identification

Houssam Akhmouch
Normandie Univ, UNICAEN
ENSICAEN, CNRS, GREYC
Crédit Agricole Brie Picardie
France

Gaël Dias
Normandie Univ, UNICAEN
ENSICAEN, CNRS, GREYC
France

Jose G. Moreno
Université de Toulouse
IRIT UMR 5505 CNRS
France

Abstract

Discovering whether words are semantically related and identifying the specific semantic relation that holds between them is of crucial importance for automatic reasoning on text data. For that purpose, different methodologies have been proposed that either (1) tackle feature engineering, (2) fine-tune latent semantic spaces, or (3) take advantage of cognitive links between semantic relations in multitask settings. In this paper, we investigate how feature engineering and multitask architectures can be improved and consequently combined to identify lexico-semantic relations. Evaluation results over a set of gold-standard datasets show that (1) combinations of similar features are beneficial (feature sets), (2) asymmetric distributional features are a strong cue to discriminate asymmetric relations as well as they play an important role in multitask architectures, (3) shared-private models improve over binary and fully-shared classifiers as well as they correctly balance the focus on features between private and shared layers[1].

1 Introduction

The ability to automatically identify lexico-semantic relations is an important issue for Information Retrieval and Natural Language Processing applications such as question answering (Dong et al., 2017), query expansion (Kathuria et al., 2017), or text summarization (Gambhir and Gupta, 2017). Lexico-semantic relations embody symmetric and asymmetric linguistic phenomena such as synonymy (e.g. phone ↔ telephone), co-hyponymy (e.g. phone ↔ monitor), hypernymy (e.g. phone → speakerphone) or meronymy (e.g. phone → mouthpiece), but more can be enumerated (Vylomova et al., 2016).

Most approaches focus on modeling a single semantic relation and consist in deciding whether a given relation r holds between a pair of words (w_1, w_2). The vast majority of efforts (Shwartz et al., 2016; Vulić and Mrkšić, 2018; Wang and He, 2020) concentrate on hypernymy which is the key organization principle of semantic memory, but studies exist on antonymy (Nguyen et al., 2017b; Ali et al., 2019), meronymy (Glavaš and Ponzetto, 2017) and co-hyponymy (Jana et al., 2020). Within this scope, different strategies have been proposed that either define new features (Santus et al., 2017; Vu and Shwartz, 2018) or build specific latent semantic spaces (Nguyen et al., 2017a; Rei et al., 2018; Wang and He, 2020) for the relation at hand.

More recently, multitask strategies have been proposed, which consist in concurrently learning correlated lexico-semantic relations (Attia et al., 2016; Balikas et al., 2019; Bannour et al., 2020), the underlying idea being that if two (or more) tasks are cognitively interlinked, a learning architecture should improve its generalization ability by taking into account the shared information existing between the tasks (Caruana, 1998).

In this paper, we propose to investigate how feature engineering can be coupled to multitask strategies for the identification of lexico-semantic relations. On the one hand, Vu and Shwartz (2018) show that the introduction of the generalized cosine ($Mult$) drastically improves results over the unique concatenation of word embeddings, thus clearly evidencing the limitations of general-purpose latent spaces. However, a complete study of symmetric and asymmetric characteristics, and their combination is still lacking, except (Santus et al., 2017), one of the most complete work in the field.

On the other hand, although existing multitask strategies have been showing promising results,

[1]Both the code and the datasets are available at https://github.com/Houssam93/Feature-Focus-in-Multi-Task-Learning-NLP for reproducibility.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2762–2772
August 1–6, 2021. ©2021 Association for Computational Linguistics

they neither take advantage of specialized features nor they implement state-of-the-art architectures, which have been successful for text classification (Liu et al., 2017). This might be due to the fact that the combination of features within shared-private multitask architectures is not straightforward, and requires specific tuning.

Evaluation results over a set of gold-standard datasets (RUMEN (Balikas et al., 2019), ROOT9 (Santus et al., 2016), WEEDS (Weeds et al., 2004) and BLESS (Baroni and Lenci, 2011)) of an architecture coupling optimized feature sets and shared-private models show that

- The combination of features within a family set improves performance over the use of a unique family member;
- Asymmetric distributional features are a strong cue to discriminate asymmetric lexico-semantic relations;
- Shared-private models improve over binary and fully-shared classifiers (Balikas et al., 2019; Bannour et al., 2020) as well as they correctly balance the focus on features between private and shared layers;
- Asymmetric distributional features play an important role in multitask architectures, being an important source of information for combining both symmetric and asymmetric tasks.

2 Related Work

Three major research directions have been proposed to identify lexico-semantic relations: (1) feature engineering, (2) construction of fine-tuned semantic spaces and (3) multitask architectures.

Within the first topic, (Levy et al., 2015) and (Vylomova et al., 2016) proposed similar evaluations to combine word input vectors ($\overrightarrow{w_1}$, $\overrightarrow{w_2}$), following initial experiments of (Baroni et al., 2012; Roller et al., 2014; Weeds et al., 2014). In particular, word pairs are encoded as the concatenation of the constituent word representations ($\overrightarrow{w_1} \oplus \overrightarrow{w_2}$), their vector difference ($\overrightarrow{w_1} - \overrightarrow{w_2}$) or their sum ($\overrightarrow{w_1} + \overrightarrow{w_2}$). Both studies evidence that the distributional hypothesis is domain-dependent by nature and as such models may not generalize across domains based on these input representations. To overcome such a limitation (Shwartz et al., 2016; Nguyen et al., 2017b) proposed to represent contextual patterns as continuous vectors with successful results, while (Vu and Shwartz, 2018) defined a generalized cosine ($\overrightarrow{w_1} \otimes \overrightarrow{w_2}$) that successfully combines with ($\overrightarrow{w_1} \oplus \overrightarrow{w_2}$).

The second main research direction aims to build fine-tuned neural latent semantic spaces. (Nguyen et al., 2017a) proposed *HyperVec*, where embeddings are learned in a specific order to capture the hypernym–hyponym distributional hierarchy from a background knowledge of hypernym-hyponym pairs. (Vulić and Mrkšić, 2018) rather proposed a post-processing strategy that retrofits the knowledge background into an original latent space. Such methods suffer from limited coverage as they affect only vectors of seen words. To deal with this limitation, (Kamath et al., 2019) presented a post-processing method that specializes vectors of all vocabulary words by learning a global specialization function, and (Wang and He, 2020) followed the same idea but proposed to learn two projection functions. In the same line, (Bouraoui et al., 2020) introduced a framework that fine-tunes BERT (Devlin et al., 2019) to include relational information.

The third approach tackles relation identification from the architecture point of view. Within this context, (Attia et al., 2016) can be viewed as a coarse-grained analysis as they propose a multitask convolutional neural network where one task acts as domain adaptation (relatedness between two words) and the second task is a multiclass classification problem for hypernymy, meronymy, synonymy and antonymy. Instead, (Balikas et al., 2019) proposed a fine-grained approach, that determines whether the learning process of a given semantic relation can be improved by the concurrent learning of another relation, where relations are synonymy, co-hyponymy, hypernymy and meronymy. (Bannour et al., 2020) implemented the same fully-shared model, but introduced the idea of data augmentation via attention models.

Although fine-tuned embeddings have evidenced improved results over generic ones, they are relation- and knowledge-dependent. One exception is proposed by (Meng et al., 2019), which learns text embeddings in a spherical space (aka. JoSE) suitable for relational information. Feature engineering also affords "cheap performance boost" (Vu and Shwartz, 2018) in resource-free environments. But, a complete study of the combination of features is still missing as well as the definition of asymmetric features in the context of continuous spaces, although a great deal of work exists for the discrete case (Kotlerman et al., 2010; Santus et al.,

2017). Finally, studies in multitask settings neither take advantage of powerful multitask models such as shared-private architectures (Liu et al., 2017) that allow to combine task-specific and cross-task information, nor benefit from the fruitful combination of distributional and pattern-based features. In this paper, we propose to deal with the aforementioned limitations in a resource-free setup.

3 Feature Engineering

Additionally to word embeddings concatenation, we define three families of features based on the distributional hypothesis (symmetric and asymmetric features) and the paradigmatic approach (pattern-based features) in continuous semantic spaces.

3.1 Distributional Representation

Most studies have been evidencing the superiority of the concatenation of representational word vectors to infer their semantic relationship (Shwartz et al., 2016; Vu and Shwartz, 2018). So, we follow this line of research. Let (w_1, w_2) be a word pair and $\overrightarrow{w_1}$, $\overrightarrow{w_2}$ their respective distributional representations of dimension d. The input distributional feature of the word pair is noted $\overrightarrow{w_1} \oplus \overrightarrow{w_2}$.

3.2 Symmetric Distributional Features

Studies have evidenced the interest of coupling word embeddings with specific features to improve relation identification. In particular, the cosine similarity measure cos has shown promising results (Garten et al., 2015; Barkan, 2017). However, Vu and Shwartz (2018) have demonstrated the effectiveness of integrating the element-wise multiplication of the input vectors, which can be seen as a generalized cosine ($cosG$, aka. $Mult$), which is defined in equation 1.

$$cosG(\overrightarrow{w_1}, \overrightarrow{w_2}) = \bigoplus_{i=1}^{d} w_1^i w_2^i \quad (1)$$

While the cosine only provides a unique value as input, $cosG$ refers to an input of dimension d, thus evidencing a dimensional issue. As a consequence, we propose to transform the $cosG$ into a unique value by using a linear activation layer as in equation 2. The $cosG1D$ can be seen as a control value of $cosG$ taking into account the dimensional bias (from high to low dimension).

$$cosG1D(\overrightarrow{w_1}, \overrightarrow{w_2}) = \sum_{i=1}^{d} \lambda_i w_1^i w_2^i \quad (2)$$

The counterpart of equation 2 is the (d times) duplication of the cosine value. This metric called cosine broadcast ($cosBr$) defined in equation 3 aims to control the dimensional issue from a low to a high dimension.

$$cosBr(\overrightarrow{w_1}, \overrightarrow{w_2}) = \bigoplus_{i=1}^{d} cos(\overrightarrow{w_1}, \overrightarrow{w_2}) \quad (3)$$

As such, in equation 4, we define a family of symmetric distributional features.

$$CosF = (cos, cosG, cosBr, cosG1D) \quad (4)$$

In the next subsection, we detail the design of new asymmetric distributional measures based on the Kullback–Leibler divergence.

3.3 Asymmetric Distributional Features

Asymmetry has shown successful results for the discrete case (Kotlerman et al., 2010; Santus et al., 2017), the underlying idea being that the relation between words may be unbalanced such that one word attracts the other one more than the opposite. Here, we define different asymmetric features in the continuous space based on the Kullback–Leibler divergence (Kullback and Leibler, 1951). To fit to the continuous case, we transform each dimension of a word vector with the sigmoid (σ) function such that all values range between 0 and 1. Thus, each word can be considered as a probability distribution and the asymmetric metric $Kull$ is defined in equations 5.

$$Kull(\overrightarrow{w_1}|\overrightarrow{w_2}) = \sum_{i=1}^{d} log(\frac{\sigma(w_1^i)}{\sigma(w_2^i)})\sigma(w_1^i) \quad (5)$$

To take into account both directions of the asymmetry, we propose to concatenate the $Kull$ values for both directions as defined in equation 6.

$$kull(\overrightarrow{w_1}, \overrightarrow{w_2}) = Kull(\overrightarrow{w_1}|\overrightarrow{w_2}) \oplus Kull(\overrightarrow{w_2}|\overrightarrow{w_1}) \quad (6)$$

Similarly to the $cosG$, we propose to define the multiplicative version of the $kull$, such that $kullG$ integrates the element-wise multiplication of the input vectors as defined in equations 7 (single asymmetry) and 8 (concatenation of both asymmetries).

$$KullG(\overrightarrow{w_1}|\overrightarrow{w_2}) = \bigoplus_{i=1}^{d} log(\frac{\sigma(w_1^i)}{\sigma(w_2^i)})\sigma(w_1^i) \quad (7)$$

$$kullG(\overrightarrow{w_1}, \overrightarrow{w_2}) = KullG(\overrightarrow{w_1}|\overrightarrow{w_2}) \oplus KullG(\overrightarrow{w_2}|\overrightarrow{w_1}) \quad (8)$$

Similarly to $cosG1D$ and to take into account the dimensional issue of the multiplicative version of the Kullback-Leibler, we define $kullG1D$ in equations 9 and 10 .

$$KullG1D(\overrightarrow{w_1}|\overrightarrow{w_2}) = \sum_{i=1}^{d} \lambda_i log(\frac{\sigma(w_1^i)}{\sigma(w_2^i)})\sigma(w_1^i) \quad (9)$$

$$kullG1D(\overrightarrow{w_1}, \overrightarrow{w_2}) = KullG1D(\overrightarrow{w_1}|\overrightarrow{w_2}) \oplus KullG1D(\overrightarrow{w_2}|\overrightarrow{w_1}) \quad (10)$$

Similarly to $cosBr$, we propose to define $kullBr$ based on the (d times) duplication of the Kulback-Leibler value for both directions as in equation 11.

$$kullBr(\overrightarrow{w_1}, \overrightarrow{w_2}) = \bigoplus_{i=1}^{d} Kull(\overrightarrow{w_1}|\overrightarrow{w_2}) \bigoplus_{i=1}^{d} Kull(\overrightarrow{w_2}|\overrightarrow{w_1}) \quad (11)$$

As such, in equation 12, we define a family of asymmetric distributional features.

$$KullF = (kull, kullG, kullBr, kullG1D) \quad (12)$$

In the next subsection, we present the encoding strategy of patterns embodying the paradigmatic approach.

3.4 Pattern-based Paradigmatic Features

Patterns are part of the paradigmatic approach (Hearst, 1992), which suggests that specific word sequences may exist that link two words in a given relation. Some examples of sequences between word pairs are given in Table 1, which evidence that some of them can be spurious, and do not necessarily include patterns.

Here, we propose to implement the methodology of (Shwartz et al., 2016) to encode patterns into continuous spaces. As such, we transform the k[2] most frequent patterns occurring between w_1 and w_2 using either BiLSTM or the Universal Sentence Encoder (USE) (Cer et al., 2018), and then perform average pooling to get the final input representation. The encoded i-th most frequent pattern is defined in equation 13, where $j \in \{\text{BiLSTM}, \text{USE}\}$, $i \in [1..k]$,

[2] k allows to deal with spurious sequences.

Relation	Path
Synonymy	**error** or **fault** √ **change** as an **alteration** √ **burning** fuel in the **combustion**
Hypernymy	**aircraft** firing rocket into an enemy **plane** √ **unit** that includes **screen** √ **act** was an unconscious **ritual**
Co-hyponymy	**pineapple** and **apricot** √ **chisel** usually used with **mallet** √ **horse** frightened by **lion** √
Meronymy	**bowl** from the world of **glass** √ **television** and **video** √ **couch** on **seat** √
Random	**reference** in the book of **mormon** **nothing** to stop the **robber** **driver** was issued traffic **ticket**

Table 1: Examples of patterns for a word pair (in bold).

and the average representation of the k patterns is noted $\overline{pat}_{*,j}^{w_1,w_2}$.

$$pat_{i,j}^{w_1,w_2} = encoder_j(w_1, path_i, w_2) \quad (13)$$

Similarly to $CosF$ and $KullF$, we define a family of pattern-based features $PatF$ in equation 14.

$$PatF = (\overline{pat}_{*,\text{USE}}, \overline{pat}_{*,\text{BiLSTM}}) \quad (14)$$

In the next section, we present the multitask settings that have been implemented to take into account relations between lexico-semantic relations.

4 Multitask Settings

Multitask architectures have shown to successfully combine closely-related lexico-semantic relations. Within this scope, the fully-shared architecture has systematically been implemented (Attia et al., 2016; Balikas et al., 2019; Bannour et al., 2020), which relies on a unique shared representation capable of solving the different tasks learned concurrently from a given input.

However, the shared-private model has proved to boost results for text classification (Liu et al., 2017). In particular, a shared-private network combines $N + 1$ different representations (one shared and N task-specific). As such, the shared layer should transfer the joint information contained in all tasks, while private layers should focus on the specific information of each task.

Moreover, $N + 1$ different input representations may coexist in the shared-private case, while a unique input representation exists for fully-shared models. Here, we propose to implement both fully-shared and shared-private architectures for different

combinations of input representations and features $X = (\overrightarrow{w_1} \oplus \overrightarrow{w_2}, CosF, KullF, PatF)$. In particular, forward selection (Kohavi and Sommerfield, 1995) is used for feature selection, as the search space is huge, 2^{10} possible combinations[3].

4.1 Multitask Architectures

The neural architectures are presented in figure 1 for two tasks. Formally, let X_k be an input vector[4], we compute a shared layer $S(X_k)$ as in equation 15, where W_{S^k} is a weight matrix, b_{S^k} a bias vector, and $k \in [1, K]$ (K the number of shared layers).

$$S(X_k) = \sigma(W_{S^k} X_k + b_{S^k}) = X_{k+1} \quad (15)$$

A private layer $H^j(Z_q)$, which solves task T_j ($j \in [1, N]$) is defined in equation 16, where $q \in [1, Q]$ (Q is the number of private layers).

$$H^j(Z_q) = \sigma(W^j_{H^q} Z_q + b^j_{H^q}) = Z_{q+1} \quad (16)$$

For the fully-shared architecture $Z_1 = S(X_K)$ and for the shared-private model $Z_1 = S(X_K) \oplus X^i$, where X^i is the specific input vector for task T_i. Finally, the N decisions are defined in equation 17.

$$O^j = \sigma(W^j_O H^j(Z_Q) + b^j_O) \quad (17)$$

The parameters are updated by minimising the binary cross-entropy. Hence, the weights of the shared layer are updated by minimising the loss function of each task alternatively, while the private layers are updated for their specific task.

4.2 Forward Selection

In order to optimize the feature combination for all $N + 1$ tasks and thus find the best input vectors for the shared and private layers (i.e. X, X^1 and X^2 in figure 1), we perform forward selection. As such, we first train the given model to find the best combination of features within a given family (i.e. within $CosF$, $KullF$ and $PatF$ individually)[5]. Once the best within-family combination has been defined for all families, we train the model for all combinations of the best within-family combinations of features. Note that for the shared-private architecture, we first train the private models independently to determine X^i ($i \in [1, N]$) and based on these findings, we train the shared-private model to determine X, constrained by the previously learned private models with input X^i.

[3] Embedding concatenation is the compulsory input.

[4] $X_1 = X$, where X is the initial input vector that combines both embeddings and a set of features specific to the task at hand.

[5] Here the model is trained three times independently for each family.

5 Experimental Setups

5.1 Datasets

There exist a large body of related works for the identification of lexico-semantic relations. The first gold-standard dataset, WEEDS, has been proposed by (Weeds et al., 2004) in the context of studies about measures of lexical similarity. Following the same objective, (Baroni and Lenci, 2011) introduced the well-known BLESS dataset, and (Santus et al., 2016) compiled the ROOT9 dataset[6], which contains word pairs randomly extracted from EVALution (Santus et al., 2015), Lenci/Benotto (Benotto, 2015) and BLESS (Baroni and Lenci, 2011). Within the context of concurrent identification of lexico-semantic relations, (Balikas et al., 2019) recently introduced the RUMEN dataset[7] to include synonymy. As the patterns are not included in the original datasets, we downloaded the English wikipedia dump[8] and extracted all patterns that do not exceed a maximum length of 10 words[9]. All datasets[10] are summarized with their specific characteristics in Table 2.

5.2 Learning Configurations

The output dimension of the Universal Sentence Encoder (USE) equals to 512. The output size of the BiLSTM $\in \{100, 200, 300, 400, 500\}$, the number of patterns ($k \in [1..5]$), the number of hidden layers ($K \in \{1, 2\}$ and $Q \in \{1, 2\}$), the number of neurons $\in \{5, 20, 50, 100, 150, 200, 300\}$ and the number of epochs ($[1..100]$) are free hyperparameters that are tuned using grid search. The weights are initialised with a uniform distribution scaled as in (Glorot and Bengio, 2010) and updated using Adam (Kingma and Ba, 2014) with a learning rate set to 0.001. The network is trained with batches of 64 examples and the number of iterations is optimized to maximize the F_1 score on the validation set. Word embeddings are initialized with the 300-dimensional representations of GloVe (Pennington

[6] https://github.com/esantus/ROOT9

[7] https://bit.ly/2Qitasd.

[8] shorturl.at/cqtQ8

[9] This value was tuned experimentally.

[10] Complete versions are available at https://github.com/Houssam93/Feature-Focus-in-Multi-Task-Learning-NLP/tree/main/Data

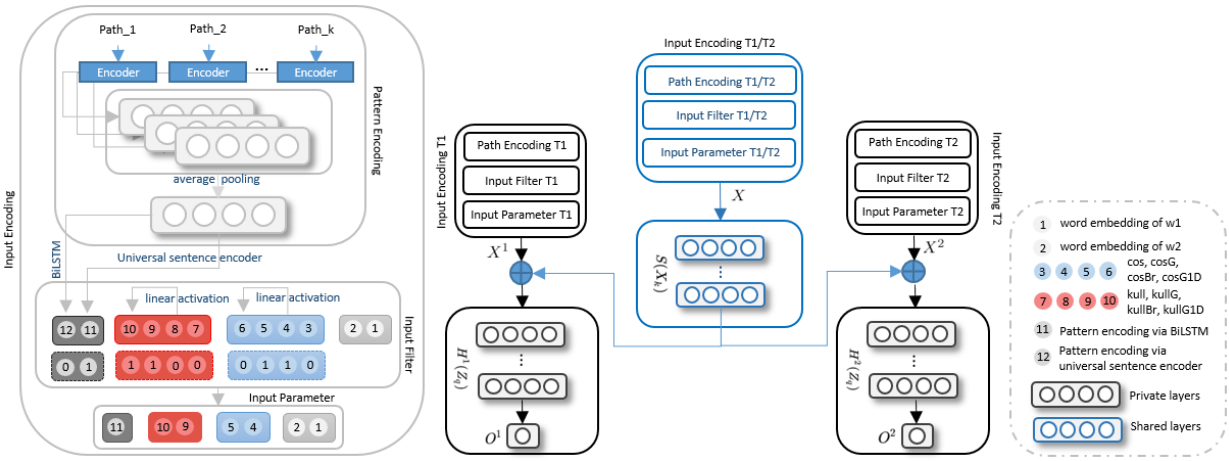

Figure 1: Fully-shared and shared-private architectures with multiple input feature combinations. The fully-shared network only includes the blue layer, i.e. $S(X_h)$.

	Synonym		Hypernym		Co-hyponym		Meronym		Random	
Dataset	#	0 / 1 / 2 / >2 (%)	#	0 / 1 / 2 / >2 (%)	#	0 / 1 / 2 / >2 (%)	#	0 / 1 / 2 / >2 (%)	#	0 / 1 / 2 / >2 (%)
RUMEN (Balikas et al., 2019)	6326	44/14/8/34	6326	65/12/5/18	-	-	-	-	6326	93/4/1/2
ROOT9 (Santus et al., 2016)	-	-	2447	21/9/6/64	3200	28/14/8/50	-	-	1100	78/9/4/9
WEEDS (Weeds et al., 2004)	-	-	1257	40/13/6/41	2083	60/11/5/24	-	-	6326	93/4/1/2
BLESS (Baroni and Lenci, 2011)	-	-	1337	57/9/4/30	3565	34/12/7/40	2943	99/0/0/1	6702	97/1/0/2

Table 2: Details of the RUMEN, ROOT9, WEEDS and BLESS datasets. 0 / 1 / 2 / >2 stands for the percentage of word pairs having respectively no pattern, 1 pattern, 2 patterns and more than 2 patterns in the Wikipedia dump.

et al., 2014) or JoSE (Meng et al., 2019). All state-of-the-art models presented in section 6 have been implemented to provide average results and perform statistical tests[11].

5.3 Lexical Split

As suggested in (Levy et al., 2015), lexical split is applied to all our experiments so that there is no vocabulary intersection between the test set and the train/validation sets. Note that for learning purposes, each dataset is split into train (50%), validation (20%) and test (30%) sub-datasets.

6 Evaluation

All comparative results against four state-of-the-art models (Shwartz et al., 2016; Vu and Shwartz, 2018; Balikas et al., 2019; Bannour et al., 2020) are presented in Table 3 for an average of 25 runs with evidenced statistical significance over four gold-standard datasets.

6.1 Private Models

We first start by analysing the impact of feature combination on private models, i.e. when a unique lexico-semantic relation is taken into account in the learning process. This stands for the first four rows of Table 3. Unsurprisingly, the introduction of a combination of (eventually new) features (*Best MLP*) outperforms existing models (Shwartz et al., 2016; Vu and Shwartz, 2018) and the multilayer perceptron (MLP) that only includes word embeddings concatenation (i.e. the simplest baseline). Note that the *Best MLP* model includes the architectures of (Shwartz et al., 2016) and (Vu and Shwartz, 2018) as it allows the combination of all family features as input.

To better understand the impact of feature engineering, we illustrate results involving all combinations of within-family features and all combinations of in-between best family features in figure 2 (a). Within the $cosF$ family alone (i.e. only cosine-based metrics are used for the learning process)[12], results clearly evidence the dimensional issue, being cos and $cosG1D$ the one-dimension metrics that evidence worst results individually. The second important finding lies in the fact that metric combination steadily improves over individual metrics. In particular, $(cosG, cosBr, cosG1D)$ gives rise to strongest results in the vast majority of cases, and particularly for hypernymy.

[11] Source codes are available at https://github.com/Houssam93/Feature-Focus-in-Multi-Task-Learning-NLP

[12] Blue dots in figure 2.

	Algorithm	Synonym vs Random				Hypernym vs Random			
		Acc.	F_1	Prec.	Rec.	Acc.	F_1	Prec.	Rec.
RUMEN	MLP	0.754	0.754	0.750	0.759	0.750	0.757	0.731	0.786
	Shwartz and Dagan (2016)	0.713	0.731	0.685	0.783	0.770	0.776	0.754	0.798
	Vu and Shwartz (2018)	0.851	0.847	0.864	0.831	0.842	0.843	0.832	0.854
	Best MLP	0.867 ⋆	0.865 ⋆	0.871 ⋆	0.859⋆†	0.863 ⋆	0.862 ⋆	0.860 ⋆	**0.865**⋆
	Balikas et al. (2019)	0.758	0.759	0.750	0.769	0.759	0.762	0.747	0.778
	Bannour et al. (2020)	0.854	0.850	0.873	0.827	0.819	0.784	0.812	0.756
	Best Fully-shared (FS)	0.861	0.864	0.843	**0.887**	0.860	0.859	0.861	0.856
	Best Shared-private (SP)	**0.870**†+	**0.866**+	**0.889** †+	0.844+	**0.869** †+	**0.867** †+	**0.871** †+	0.864+
		Co-hyponym vs Random				Hypernym vs Random			
	Algorithm	Acc.	F_1	Prec.	Rec.	Acc.	F_1	Prec.	Rec.
ROOT9	MLP	0.909	0.939	0.954	0.925	0.904	0.936	0.944	0.929
	Shwartz and Dagan (2016)	0.919	0.946	0.955	0.938	0.842	0.901	0.860	0.946
	Vu and Shwartz (2018)	0.940	0.961	0.962	0.959	0.943	0.962	0.961	0.964⋆
	Best MLP	0.950 ⋆	0.967 ⋆	**0.973**⋆	0.959	**0.947**⋆†	**0.965**⋆†	**0.971**†⋆	0.959†
	Balikas et al. (2019)	0.909	0.940	0.949	0.931	0.911	0.941	0.949	0.932
	Bannour et al. (2020)	0.949	0.966	0.964	**0.969**+	0.908	0.932	0.941	0.923
	Best Fully-shared (FS)	0.947	0.965	0.971	0.959	0.944	0.963	0.964	**0.962**
	Best Shared-private (SP)	**0.951**+	**0.968** +	0.971+	0.964†	0.943+	0.962+	0.969+	0.955+
		Co-hyponym vs Random				Hypernym vs Random			
	Algorithm	Acc.	F_1	Prec.	Rec.	Acc.	F_1	Prec.	Rec.
WEEDS	MLP	0.720	0.449	0.422	0.479	0.726	0.457	0.432	0.485
	Shwartz and Dagan (2016)	0.769	0.532	0.513	0.552	0.716	0.474	0.423	0.539
	Vu and Shwartz (2018)	0.848	0.691	0.669	0.714	0.833	0.661	0.641	0.682
	Best MLP	0.873 ⋆	0.737⋆	0.729⋆	0.746⋆†	0.886 ⋆	0.746⋆	0.797⋆	0.701⋆
	Balikas et al. (2019)	0.721	0.443	0.422	0.466	0.724	0.462	0.431	0.498
	Bannour et al. (2020)	0.871	0.713	0.754	0.678	**0.924**+	**0.751** +	**0.854**+	0.669
	Best Fully-shared (FS)	0.864	0.737	0.685	**0.796**	0.873	0.736	0.727	**0.746**
	Best Shared-private (SP)	**0.890**†+	**0.761**†+	**0.789**†+	0.736+	0.882+	0.743	0.771	0.717†
		Meronym vs Random				Hypernym vs Random			
	Algorithm	Acc.	F_1	Prec.	Rec.	Acc.	F_1	Prec.	Rec.
BLESS	MLP	0.839	0.748	0.805	0.698	0.845	0.762	0.797	0.731
	Shwartz and Dagan (2016)	0.855	0.781	0.807	0.756	0.842	0.754	0.804	0.709
	Vu and Shwartz (2018)	0.886	0.820	0.883	0.765	0.882	0.811	0.891⋆	0.744
	Best MLP	0.909 ⋆	0.864 ⋆	0.883	**0.847** ⋆	0.905⋆	0.859 ⋆	0.872	0.847 ⋆†
	Balikas et al. (2019)	0.846	0.759	0.814	0.711	0.848	0.764	0.812	0.721
	Bannour et al. (2020)	0.896	0.837	**0.913**+	0.770	**0.954** +	0.821	0.883	0.769
	Best Fully-shared (FS)	0.903	0.850	0.895	0.810	0.906	0.862	0.861	**0.864**
	Best Shared-private (SP)	**0.912** †+	**0.868**†+	0.890†	**0.847**+	0.916	**0.873**†+	**0.906**†+	0.843+

Table 3: Overall results for all architectures with GloVe embeddings. Lexical split is applied. ⋆, † and + denote p-value ≤ 0.05 based on the t-Test assuming unequal sample variances of metric values between respectively (Best MLP) against (Vu and Shwartz, 2018), (Best SP) against (Best MLP), and (Best SP) against (Bannour et al., 2020).

Within the $KullF$ family alone[13], results seem to indicate that $kullBr$ is the less performing (alone and in combination) feature, although regularities are difficult to establish as different results can be observed depending on the dataset. Similarly to the previous observation, the combination of asymmetric features provides improved results for the vast majority of cases, suggesting that individual values encode complementary information.

Within the $PatF$ family[14], the BiLSTM encoding seems to provide superior results to the USE encoding, but more importantly, results clearly show that pattern-based features can be a strong cue for the classification process provided that a large number of patterns can be extracted, as it is shown for ROOT9 (see Table 2 for the number of patterns).

More surprisingly, the $CosF$ features steadily indicate stronger results than the $KullF$ and $PatF$ features for asymmetric relations (hypernymy and meronymy), thus suggesting that symmetry is an important characteristic for all relations.

Finally, results clearly show that the combinations of the best features per family[15] steadily outperform results of individual family features, thus demonstrating their complementarity. In particular, symmetric and asymmetric distributional features successfully combine for asymmetric relations, and the successful combination is with pattern-based and cosine-based features for co-hyponymy. However, only symmetric distributional features allow maximum performance for synonymy, which can easily be understood as this is a symmetric relation. To strengthen our comments, we give the distribution of features for the best configurations in Table 4 (first row) for all datasets and relations.

6.2 Multitask Models

Results of the multitask architectures are presented in rows 5-8 of Table 3. In particular, the *Best Fully-shared* network stands for the model of Balikas et al. (2019) with an optimized set of input features, oppositely to their settings which rely on the unique concatenation of word embeddings. Figures clearly

[13] Red dots in figure 2.

[14] Black dots in figure 2.

[15] Green dots in figure 2.

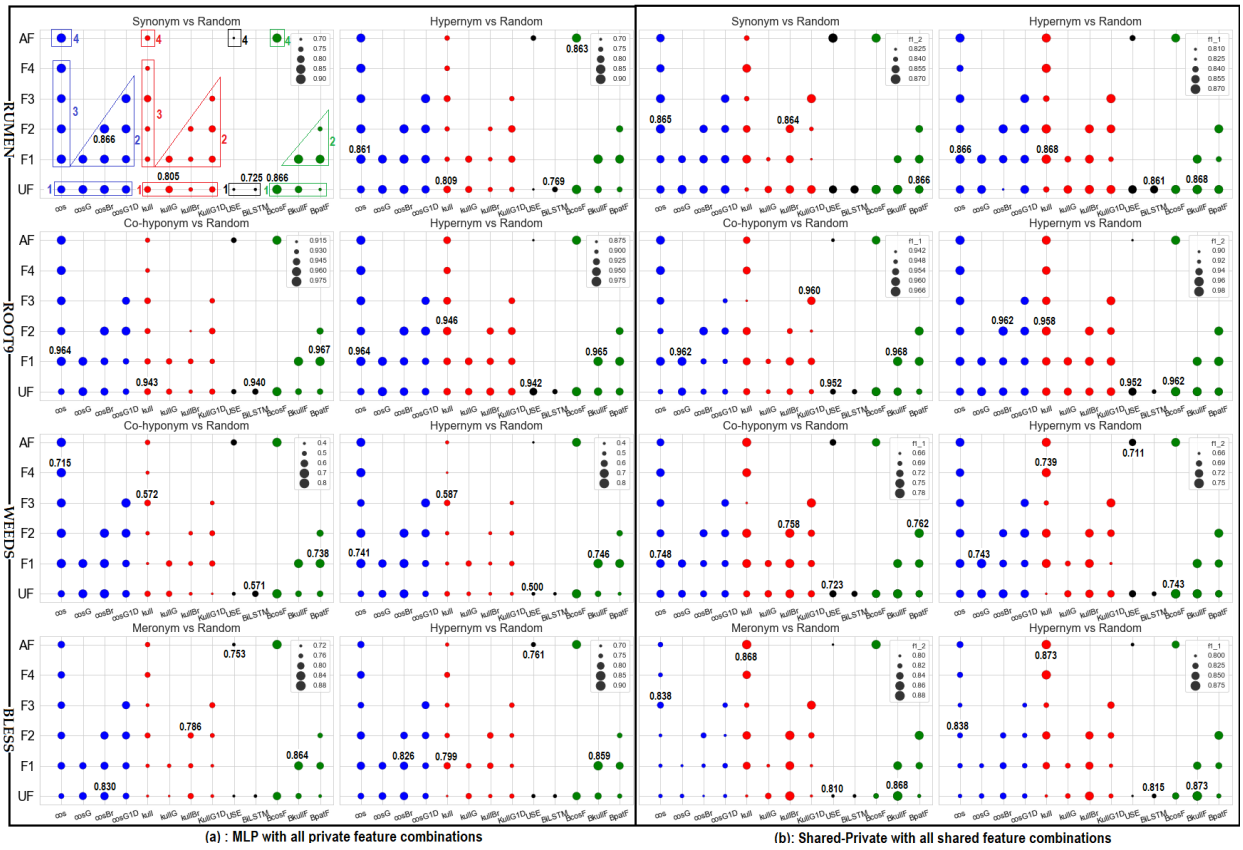

Figure 2: F_1 score results for all feature combinations. Bounding box 1 stands for any individual feature alone, e.g. ($cosG$, UF) means only $cosG$. Bounding box 2 stands for any 2-by-2 combination of features, e.g. ($cosBr$,F2) refers to ($cosBr$,$cosG$). Bounding box 3 refers to the ablation of one feature from the set of all features, e.g. (Bounding box 3 Blue, F2) refers to (cos,$cosBr$,$cosG1D$). Bounding box 4 stands for the combination of all features for a given family (AF). $BcosF$, $BKullF$ and $BpatF$ stand for best combination of features within its respective family.

		Synonym vs Random			Hypernym vs Random		
	Algorithm	CosF	KullF	PatF	CosF	KullF	PatF
RUMEN	Best MLP	0 1 1 0	0 0 0 0	0 0	0 1 1 1	1 0 0 0	0 1
	Best FS	0 1 1 0	0 0 0 0	0 0	1 1 1 1	0 0 0 0	0 0
	Best SP	0 0 0 0	0 0 0 0	1 1	0 0 0 0	0 1 1 1	0 0
		Co-hyponymy vs Random			Hypernym vs Random		
	Algorithm	CosF	KullF	PatF	CosF	KullF	PatF
ROOT9	Best MLP	0 1 1 1	0 0 0 0	1 1	0 1 1 1	1 0 1 1	0 0
	Best FS	0 1 1 0	0 0 0 0	0 0	1 1 1 1	0 0 0 0	0 0
	Best SP	0 1 1 0	0 1 1 0	0 0	1 1 1 1	0 0 0 0	1 0
		Co-hyponymy vs Random			Hypernym vs Random		
	Algorithm	CosF	KullF	PatF	CosF	KullF	PatF
WEEDS	Best MLP	1 1 1 0	0 0 0 0	0 1	0 1 1 1	1 1 0 1	0 0
	Best FS	1 1 1 0	0 0 0 0	0 0	0 1 1 1	0 0 0 0	0 0
	Best SP	0 0 0 0	0 1 1 0	1 0	1 1 0 0	0 0 0 0	0 0
		Meronym vs Random			Hypernym vs Random		
	Algorithm	CosF	KullF	PatF	CosF	KullF	PatF
BLESS	Best MLP	0 0 1 0	0 1 1 0	0 0	1 0 1 0	0 1 1 1	0 0
	Best FS	0 0 1 0	1 0 1 1	0 1	0 0 1 0	0 0 0 0	0 1
	Best SP	0 0 0 0	1 1 1 1	0 0	0 0 0 0	1 1 1 1	0 0

Table 4: Best combinations of features for all models. 0 and 1 stand for the absence or the presence, respectively, of the given feature within its family, where the order is given by equations 4, 12 and 14.

show the superiority of the shared-private network (*Best SP*) over the fully-shared model (*Best FS*) for most cases, suggesting that the combination of private and shared information is beneficial to the decision process. However, the *Best MLP* is a hard model to beat as the *Best SP* statistically outperforms the former architecture 4 times out of 8, and 2 times out of 8 without statistical significance. But the contrary is only true for ROOT9 (wrt. F_1 score), where *Best MLP* statistically exceeds *Best SP*.

The important issue in shared-private architectures is to understand how well they distribute the feature space between private and shared layers. For that purpose, we analyse figure 2 (b), which shows feature combinations for the shared layer, i.e. when two tasks are learned concurrently. Note that in this case, best combinations from the private models (learned separately) restrict the learning process. The first main conclusion is that asymmetric distributional features ($KullF$) steadily compete with cosine-based features ($CosF$), even clearly outperforming the latter for BLESS, which is definitely not the case within private models. The same conclusion can be drawn

		Synonym vs Random		Hypernym vs Random	
	Algorithm	GloVe	JoSE	GloVe	JoSE
RUMEN	MLP	0.754	0.730	0.757	0.731
	Best MLP	0.865	**0.870**	0.862	0.863
	Best SP	0.866	0.869⋆	**0.867**⋆	0.865†
		Co-hyponym vs Random		Hypernym vs Random	
	Algorithm	GloVe	JoSE	GloVe	JoSE
ROOT9	MLP	0.939	0.927	0.936	0.922
	Best MLP	0.967	0.967	0.965	0.963
	Best SP	**0.968**⋆	0.966	0.962	**0.966**⋆†
		Co-hyponym vs Random		Hypernym vs Random	
	Algorithm	GloVe	JoSE	GloVe	JoSE
WEEDS	MLP	0.449	0.458	0.457	0.455
	Best MLP	0.737	0.758	0.746	0.754
	Best SP	0.761	**0.764**⋆†	0.743	**0.759**⋆†
		Meronym vs Random		Hypernym vs Random	
	Algorithm	GloVe	JoSE	GloVe	JoSE
BLESS	MLP	0.748	0.810	0.762	0.798
	Best MLP	0.864	0.865	0.859	0.850
	Best SP	**0.868**⋆	0.865	0.873	**0.874**†

Table 5: F_1 scores with GloVe and JoSE. ⋆ and † denote p-value ≤ 0.05 based on the t-Test assuming unequal sample variances of metric values between respectively (Best SP JoSE) vs. (Best SP GloVe) and (Best SP JoSE) vs. (Best MLP JoSE).

for pattern-based features $PatF$, which impact is much more important in the shared layers than it is the case in the private models when compared to $CosF$. This suggests that when private models focus more on symmetric features, shared-private models take advantage of asymmetric features to capture task dissimilarity (indeed in the concurrent tasks there is always at least one asymmetric task).

Another interesting observation is that best models are usually not a combination of different family features. Only 2 cases out of 8 show improved results with feature combination. In fact, such results suggest that private and shared layers distinctively balance the family feature space. We clearly see this situation in Table 4 by looking at the complementarity of the input feature vectors of private (row 1) and shared-private models (row 3). For instance, when maximizing the hypernymy task within the shared-private model over RUMEN, the private input vectors are $(cosG, cosBr, cosG1D, kull, \overline{pat}_{*,\text{BiLSTM}})$ for hypernymy and $(cosG, cosBr)$ for synonymy, while the shared input vector is $(kullG, kullBr, kullG1D)$. It is worth noticing that this situation does not hold for the fully-shared models as they are clearly biased towards cosine-based metrics and rarely include asymmetric distributional and pattern-based features.

6.3 Spherical text embeddings

We propose to compare our feature-based architectures with relational embeddings, namely JoSE (Meng et al., 2019), the underlying idea being to understand how feature-based strategies can compare and eventually add-on to fine-tuned neural semantic spaces. Results are illustrated in Table 5.

Results of the baseline *MLP* model do not evidence a clear advantage of relational embeddings compared to general-purpose ones like GloVe, BLESS being the only exception. However, it is interesting to notice that the proportion of improvement is much more important for JoSE embeddings when introducing combinations of features. Indeed, while the *MLP* model with GloVe overtakes the JoSE version 5 times out of 8, the *Best MLP* model with JoSE overtakes the GloVe version 5 times out of 8, thus suggesting that spherical embeddings are sensitive to feature engineering.

Finally, while shared-private architectures provide overall best results, a clear distinction between both embeddings is difficult to establish, although a small tendency towards JoSE embeddings seems to emerge. Indeed, while the hypernymy relation is better tackled by relational embeddings (3 out of 4 configurations), meronymy is better handled by GloVe although being an asymmetric relation. With respect to symmetric relations (synonymy and co-hyponymy), the situation slightly converges towards relational embeddings with better results in 2 out of 3 experiments.

7 Conclusions

In this paper, we proposed the definition of asymmetric distributional features in continuous spaces based on the Kullback-Leibler divergence, and suggested to combine them with families of symmetric distributional and pattern-based characteristics using a feature selection process. We proposed to analyse the impact of feature combination in multi-task settings, which combine private and shared layers. Results evidenced the benefits of feature combination in the private models, and they highlighted the importance of asymmetric (distributional and paradigmatic) features in the shared layers. Moreover, share-private architectures showed the capacity of balancing feature families between private and shared layers thus taking full advantage of most features in the decision process.

References

Muhammad Asif Ali, Yifang Sun, Xiaoling Zhou, Wei Wang, and Xiang Zhao. 2019. Antonym-synonym classification based on new sub-space embeddings.

In *33rd AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 6204–6211.

Mohammed Attia, Suraj Maharjan, Younes Samih, Laura Kallmeyer, and Thamar Solorio. 2016. Cogalex-v shared task: Ghhh - detecting semantic relations via word embeddings. In *Workshop on Cognitive Aspects of the Lexicon*, pages 86–91.

Georgios Balikas, Gaël Dias, Rumen Moraliyski, Houssam Akhmouch, and Massih-Reza Amini. 2019. Learning lexical-semantic relations using intuitive cognitive links. In *41st European Conference on Information Retrieval (ECIR)*, pages 3–18.

Nesrine Bannour, Gaël Dias, Youssef Chahir, and Houssam Akhmouch. 2020. Patch-based identification of lexical semantic relations. In *42nd European Conference on Information Retrieval (ECIR)*, pages 126–140.

Oren Barkan. 2017. Bayesian neural word embedding. In *31st AAAI Conference on Artificial Intelligence*, volume 31.

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

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics (GEMS)*, pages 1–10.

Giulia Benotto. 2015. *Distributional Models for Semantic Relations: A Sudy on Hyponymy and Antonymy*. Ph.D. thesis, University of Pisa.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *34th AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 7456–7463.

Rich Caruana. 1998. Multitask learning. In *Learning to learn*, pages 95–133. Springer.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, page 4171–4186.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 886–897.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66.

Justin Garten, Kenji Sagae, Volkan Ustun, and Morteza Dehghani. 2015. Combining distributed vector representations for words. In *1st workshop on vector space modeling for natural language processing*, pages 95–101.

Goran Glavaš and Simone Paolo Ponzetto. 2017. Dual tensor model for detecting asymmetric lexico-semantic relations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1758–1768.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *13th International Conference on Artificial Intelligence and Statistics (AISTAT)*, pages 249–256.

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *14th Conference on Computational Linguistics (COLING)*, pages 539–545.

Abhik Jana, Nikhil Reddy Varimalla, and Pawan Goyal. 2020. Using distributional thesaurus embedding for co-hyponymy detection. In *12th Language Resources and Evaluation Conference (LREC)*, pages 5766–5771.

Aishwarya Kamath, Jonas Pfeiffer, Edoardo Maria Ponti, Goran Glavaš, and Ivan Vulić. 2019. Specializing distributional vectors of all words for lexical entailment. In *4th Workshop on Representation Learning for NLP (RepL4NLP)*, pages 72–83.

Neha Kathuria, Kanika Mittal, and Anusha Chhabra. 2017. A comprehensive survey on query expansion techniques, their issues and challenges. *International Journal of Computer Applications*, 168(12).

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ron Kohavi and Dan Sommerfield. 1995. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In *1st International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 192–197.

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

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

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

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance M. Kaplan, and Jiawei Han. 2019. Spherical text embedding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8206–8215.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017a. Hierarchical embeddings for hypernymy detection and directionality. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 233–243.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017b. Distinguishing antonyms and synonyms in a pattern-based neural network. In *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 76–85.

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

Marek Rei, Daniela Gerz, and Ivan Vulić. 2018. Scoring lexical entailment with a supervised directional similarity network. In *56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 638–643.

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

Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016. Nine features in a random forest to learn taxonomical semantic relations. In *10th International Conference on Language Resources and Evaluation (LREC)*, pages 4557–4564.

Enrico Santus, Vered Shwartz, and Dominik Schlechtweg. 2017. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 65–75.

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 *4th Workshop on Linked Data in Linguistics (LDL) associated to Association for Computational Linguistics and Asian Federation of Natural Language Processing (ACL-IJCNLP)*, pages 64–69.

Vered Shwartz and Ido Dagan. 2016. Path-based vs. distributional information in recognizing lexical semantic relations. In *5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pages 24–29.

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

Tu Vu and Vered Shwartz. 2018. Integrating multiplicative features into supervised distributional methods for lexical entailment. In *7th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 160–166.

Ivan Vulić and Nikola Mrkšić. 2018. Specialising word vectors for lexical entailment. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1134–1145.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1671–1682.

Chengyu Wang and Xiaofeng He. 2020. BiRRE: Learning bidirectional residual relation embeddings for supervised hypernymy detection. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3630–3640.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David J. Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *5th International Conference on Computational Linguistics (COLING)*, pages 2249–2259.

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