

# Post-Specialisation: Retrofitting Vectors of Words Unseen in Lexical Resources

Ivan Vulić<sup>1</sup>, Goran Glavaš<sup>2</sup>, Nikola Mrkšić<sup>3</sup>, Anna Korhonen<sup>1</sup>

<sup>1</sup> Language Technology Lab, University of Cambridge

<sup>2</sup> Data and Web Science Group, University of Mannheim

<sup>3</sup> PolyAI

{iv250, alk23}@cam.ac.uk

goran@informatik.uni-mannheim.de nikola@poly-ai.com

## Abstract

Word vector *specialisation* (also known as *retrofitting*) is a portable, light-weight approach to fine-tuning arbitrary distributional word vector spaces by *injecting* external knowledge from rich lexical resources such as WordNet. By design, these *post-processing* methods only update the vectors of words occurring in external lexicons, leaving the representations of all *unseen* words intact. In this paper, we show that constraint-driven vector space specialisation can be extended to unseen words. We propose a novel *post-specialisation* method that: **a)** preserves the useful linguistic knowledge for seen words; while **b)** propagating this external signal to unseen words in order to improve their vector representations as well. Our post-specialisation approach explicits a non-linear specialisation function in the form of a deep neural network by learning to predict specialised vectors from their original distributional counterparts. The learned function is then used to specialise vectors of unseen words. This approach, applicable to any post-processing model, yields considerable gains over the initial specialisation models both in intrinsic word similarity tasks, and in two downstream tasks: *dialogue state tracking* and *lexical text simplification*. The positive effects persist across three languages, demonstrating the importance of specialising the full vocabulary of distributional word vector spaces.

## 1 Introduction

Word representation learning is a key research area in current Natural Language Processing (NLP), with its usefulness demonstrated across a range of tasks (Collobert et al., 2011; Chen and Manning, 2014; Melamud et al., 2016b). The standard techniques for inducing distributed word representations are grounded in the distributional hypothesis (Harris, 1954): they rely on co-occurrence information in large textual corpora (Mikolov et al., 2013b;

Pennington et al., 2014; Levy and Goldberg, 2014; Levy et al., 2015; Bojanowski et al., 2017). As a result, these models tend to coalesce the notions of semantic similarity and (broader) conceptual relatedness, and cannot accurately distinguish antonyms from synonyms (Hill et al., 2015; Schwartz et al., 2015). Recently, we have witnessed a rise of interest in representation models that move beyond stand-alone unsupervised learning: they leverage external knowledge in human- and automatically-constructed lexical resources to enrich the semantic content of distributional word vectors, in a process termed *semantic specialisation*.

This is often done as a post-processing (sometimes referred to as *retrofitting*) step: input word vectors are *fine-tuned* to satisfy linguistic constraints extracted from lexical resources such as WordNet or BabelNet (Faruqui et al., 2015; Mrkšić et al., 2017). The use of external curated knowledge yields improved word vectors for the benefit of downstream applications (Faruqui, 2016). At the same time, this specialisation of the distributional space distinguishes between true similarity and relatedness, and supports language understanding tasks (Kiela et al., 2015; Mrkšić et al., 2017).

While there is consensus regarding their benefits and ease of use, one property of the post-processing specialisation methods slips under the radar: most existing post-processors update word embeddings only for words which are present (i.e., *seen*) in the external constraints, while vectors of all other (i.e., *unseen*) words remain unaffected. In this work, we propose a new approach that extends the specialisation framework to unseen words, relying on the transformation of the vector (sub)space of seen words. Our intuition is that the process of fine-tuning seen words provides implicit information on how to leverage the external knowledge to unseen words. The method should preserve the already injected knowledge for seen words, simultaneously

propagating the external signal to unseen words in order to improve their vectors.

The proposed *post-specialisation* method can be seen as a two-step process, illustrated in Fig. 1a: **1)** We use a state-of-the-art specialisation model to transform the subspace of seen words from the input distributional space into the specialised subspace; **2)** We learn a mapping function based on the transformation of the “seen subspace”, and then apply it to the distributional subspace of unseen words. We allow the proposed post-specialisation model to learn from large external linguistic resources by implementing the mapping as a deep feed-forward neural network with non-linear activations. This allows the model to learn the generalisation of the fine-tuning steps taken by the initial specialisation model, itself based on a very large number (e.g., hundreds of thousands) of external linguistic constraints.

As indicated by the results on word similarity and two downstream tasks (dialogue state tracking and lexical text simplification) our post-specialisation method consistently outperforms state-of-the-art methods which specialise seen words only. We report improvements using three distinct input vector spaces for English and for three test languages (English, German, Italian), verifying the robustness of our approach.

## 2 Related Work and Motivation

**Vector Space Specialisation** A standard approach to incorporating external and background knowledge into word vector spaces is to pull the representations of similar words closer together and to push words in undesirable relations (e.g., antonyms) away from each other. Some models integrate such constraints into the training procedure and *jointly* optimize distributional and non-distributional objectives: they modify the prior or the regularisation (Yu and Dredze, 2014; Xu et al., 2014; Bian et al., 2014; Kiela et al., 2015), or use a variant of the SGNS-style objective (Liu et al., 2015; Ono et al., 2015; Osborne et al., 2016; Nguyen et al., 2017). In theory, word embeddings obtained by these joint models could be as good as representations produced by models which fine-tune input vector space. However, their performance falls behind that of fine-tuning methods (Wieting et al., 2015). Another disadvantage is that their architecture is tied to a specific underlying model (typically `word2vec` models).

In contrast, fine-tuning models inject external knowledge from available lexical resources (e.g., WordNet, PPDB) into pre-trained word vectors as a *post-processing step* (Faruqui et al., 2015; Rothe and Schütze, 2015; Wieting et al., 2015; Nguyen et al., 2016; Mrkšić et al., 2016; Cotterell et al., 2016; Mrkšić et al., 2017). Such post-processing models are popular because they offer a portable, flexible, and light-weight approach to incorporating external knowledge into *arbitrary* vector spaces, yielding state-of-the-art results on language understanding tasks (Faruqui et al., 2015; Mrkšić et al., 2016; Kim et al., 2016; Vulić et al., 2017b).

Existing post-processing models, however, suffer from a major limitation. Their *modus operandi* is to enrich the distributional information with external knowledge only if such knowledge is present in a lexical resource. This means that they update and improve only representations of words actually seen in external resources. Because such words constitute only a fraction of the whole vocabulary (see Sect. 4), most words, unseen in the constraints, retain their original vectors. The main goal of this work is to address this shortcoming by specialising *all* words from the initial distributional space.

## 3 Methodology: Post-Specialisation

Our starting point is the state-of-the-art specialisation model ATTRACT-REPEL (AR) (Mrkšić et al., 2017), outlined in Sect. 3.1. We opt for the AR model due to its strong performance and ease of use, but we note that the proposed post-specialisation approach for specialising unseen words, described in Sect. 3.2, is applicable to any post-processor, as empirically validated in Sect. 5.

### 3.1 Initial Specialisation Model: AR

Let  $\mathcal{V}_s$  be the vocabulary,  $A$  the set of synonymous ATTRACT word pairs (e.g., *rich* and *wealthy*), and  $R$  the set of antonymous REPEL word pairs (e.g., *increase* and *decrease*). The ATTRACT-REPEL procedure operates over mini-batches of such pairs  $\mathcal{B}_A$  and  $\mathcal{B}_R$ . Let each word pair  $(x_l, x_r)$  in these sets correspond to a vector pair  $(\mathbf{x}_l, \mathbf{x}_r)$ . A mini-batch of  $b_{att}$  attract word pairs is given by  $\mathcal{B}_A = [(\mathbf{x}_l^1, \mathbf{x}_r^1), \dots, (\mathbf{x}_l^{k_1}, \mathbf{x}_r^{k_1})]$  (analogously for  $\mathcal{B}_R$ , which consists of  $b_{rep}$  pairs).

Next, the sets of negative examples  $T_A = [(\mathbf{t}_l^1, \mathbf{t}_r^1), \dots, (\mathbf{t}_l^{k_1}, \mathbf{t}_r^{k_1})]$  and  $T_R = [(\mathbf{t}_l^1, \mathbf{t}_r^1), \dots, (\mathbf{t}_l^{k_2}, \mathbf{t}_r^{k_2})]$  are defined as pairs of *negative examples* for each  $A$  and  $R$

pair in mini-batches  $\mathcal{B}_A$  and  $\mathcal{B}_R$ . These negative examples are chosen from the word vectors present in  $\mathcal{B}_A$  or  $\mathcal{B}_R$  so that, for each  $A$  pair  $(\mathbf{x}_l, \mathbf{x}_r)$ , the negative example pair  $(\mathbf{t}_l, \mathbf{t}_r)$  is chosen so that  $\mathbf{t}_l$  is the vector closest (in terms of cosine distance) to  $\mathbf{x}_l$  and  $\mathbf{t}_r$  is closest to  $\mathbf{x}_r$ .<sup>1</sup> The negatives are used **1**) to force  $A$  pairs to be closer to each other than to their respective negative examples; and **2**) to force  $R$  pairs to be further away from each other than from their negative examples. The first term of the cost function pulls  $A$  pairs together:

$$\text{Att}(\mathcal{B}_A, \mathcal{B}_R) = \sum_{i=1}^{b_{att}} \left[ \tau \left( \delta_{att} + \mathbf{x}_l^i \mathbf{t}_l^i - \mathbf{x}_l^i \mathbf{x}_r^i \right) + \tau \left( \delta_{att} + \mathbf{x}_r^i \mathbf{t}_r^i - \mathbf{x}_l^i \mathbf{x}_r^i \right) \right] \quad (1)$$

where  $\tau(z) = \max(0, z)$  is the standard rectifier function (Nair and Hinton, 2010) and  $\delta_{att}$  is the attract margin: it determines how much closer these vectors should be to each other than to their respective negative examples. The second, REPEL term in the cost function is analogous: it pushes  $R$  word pairs away from each other by the margin  $\delta_{rep}$ .

Finally, in addition to the  $A$  and  $R$  terms, a regularisation term is used to preserve the semantic content originally present in the distributional vector space, as long as this information does not contradict the injected external knowledge. Let  $\mathcal{V}(\mathcal{B})$  be the set of all word vectors present in a mini-batch, the distributional regularisation term is then:

$$\text{Reg}(\mathcal{B}_A, \mathcal{B}_R) = \sum_{\mathbf{x}_i \in \mathcal{V}(\mathcal{B}_A \cup \mathcal{B}_R)} \lambda_{reg} \|\widehat{\mathbf{x}}_i - \mathbf{x}_i\|_2 \quad (2)$$

where  $\lambda_{reg}$  is the  $L_2$ -regularisation constant and  $\widehat{\mathbf{x}}_i$  denotes the original (distributional) word vector for word  $x_i$ . The full ATTRACT-REPEL cost function is finally constructed as the sum of all three terms.

### 3.2 Specialisation of Unseen Words

**Problem Formulation** The goal is to learn a *global transformation function* that generalises the perturbations of the initial vector space made by ATTRACT-REPEL (or any other specialisation procedure), as conditioned on the external constraints. The learned function propagates the signal coded in the input constraints to all the words unseen during the specialisation process. We seek a regression

<sup>1</sup>Similarly, for each  $R$  pair  $(\mathbf{x}_l, \mathbf{x}_r)$ , the negative pair  $(\mathbf{t}_l, \mathbf{t}_r)$  is chosen from the in-batch vectors so that  $\mathbf{t}_l$  is the vector furthest away from  $\mathbf{x}_l$  and  $\mathbf{t}_r$  is furthest from  $\mathbf{x}_r$ . All vectors are unit length (re)normalised after each epoch.

function  $f : \mathbb{R}^{dim} \rightarrow \mathbb{R}^{dim}$ , where  $dim$  is the vector space dimensionality. It maps word vectors from the initial vector space  $\mathbf{X}$  to the specialised target space  $\mathbf{X}'$ . Let  $\widehat{\mathbf{X}}' = f(\mathbf{X})$  refer to the predicted mapping of the vector space, while the mapping of a single word vector is denoted  $\widehat{\mathbf{x}}'_i = f(\mathbf{x}_i)$ .

An input distributional vector space  $\mathbf{X}_d$  represents words from a vocabulary  $\mathcal{V}_d$ .  $\mathcal{V}_d$  may be divided into two vocabulary subsets:  $\mathcal{V}_d = \mathcal{V}_s \cup \mathcal{V}_u$ ,  $\mathcal{V}_s \cap \mathcal{V}_u = \emptyset$ , with the accompanying vector subspaces  $\mathbf{X}_d = \mathbf{X}_s \sqcup \mathbf{X}_u$ .  $\mathcal{V}_s$  refers to the vocabulary of seen words: those that appear in the external linguistic constraints and have their embeddings changed in the specialisation process.  $\mathcal{V}_u$  denotes the vocabulary of unseen words: those not present in the constraints and whose embeddings are unaffected by the specialisation procedure.

The AR specialisation process transforms only the subspace  $\mathbf{X}_s$  into the specialised subspace  $\mathbf{X}'_s$ . All words  $x_i \in \mathcal{V}_s$  may now be used as training examples for learning the explicit mapping function  $f$  from  $\mathbf{X}_s$  into  $\mathbf{X}'_s$ . If  $N = |\mathcal{V}_s|$ , we in fact rely on  $N$  training pairs:  $(\mathbf{x}_i, \mathbf{x}'_i) = \{\mathbf{x}_i \in \mathbf{X}_s, \mathbf{x}'_i \in \mathbf{X}'_s\}$ . Function  $f$  can then be applied to unseen words  $x \in \mathcal{V}_u$  to yield the specialised subspace  $\widehat{\mathbf{X}}'_u = f(\mathbf{X}_u)$ . The specialised space containing *all* words is then  $\mathbf{X}_f = \mathbf{X}'_s \cup \widehat{\mathbf{X}}'_u$ . The complete high-level post-specialisation procedure is outlined in Fig. 1a.

Note that another variant of the approach could obtain  $\mathbf{X}_f$  as  $\mathbf{X}_f = f(\mathbf{X}_d)$ , that is, the entire distributional space is transformed by  $f$ . However, this variant seems counter-intuitive as it forgets the actual output of the initial specialisation procedure and replaces word vectors from  $\mathbf{X}'_s$  with their approximations, i.e.,  $f$ -mapped vectors.<sup>2</sup>

**Objective Functions** As mentioned, the  $N$  seen words  $x_i \in \mathcal{V}_s$  in fact serve as our “pseudo-translation” pairs supporting the learning of a cross-space mapping function. In practice, in its high-level formulation, our mapping problem is equivalent to those encountered in the literature on cross-lingual word embeddings where the goal is to learn a shared cross-lingual space given monolingual vector spaces in two languages and  $N_1$  translation pairs (Mikolov et al., 2013a; Lazaridou et al., 2015; Vulić and Korhonen, 2016b; Artetxe et al., 2016, 2017; Conneau et al., 2017; Ruder et al., 2017). In our setup, the standard objective based on  $L_2$ -penalised

<sup>2</sup>We have empirically confirmed the intuition that the first variant is superior to this alternative. We do not report the actual quantitative comparison for brevity.

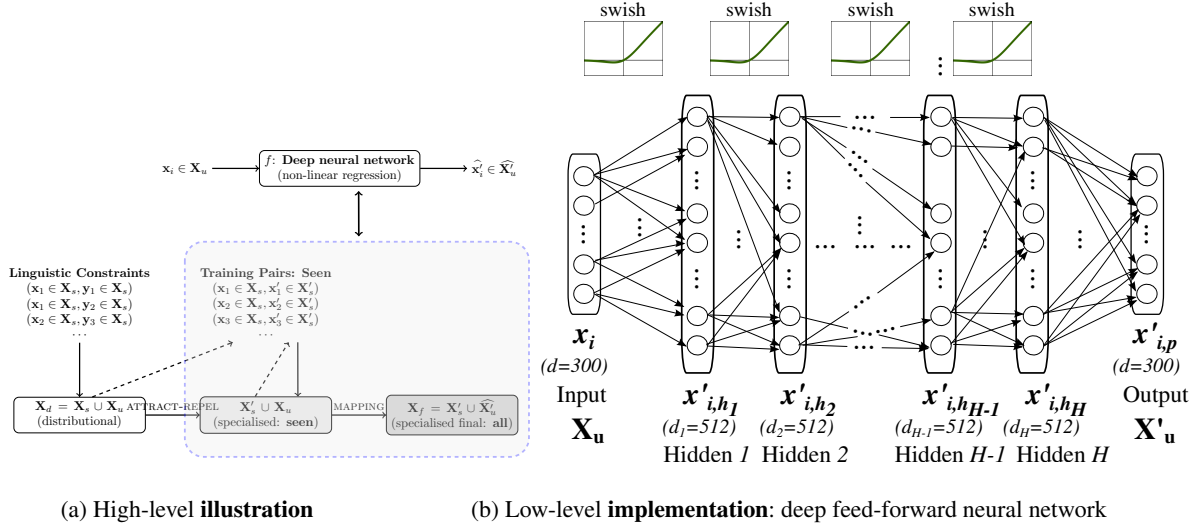


Figure 1: **(a)** High-level illustration of the post-specialisation approach: the subspace  $\mathbf{X}_s$  of the initial distributional vector space  $\mathbf{X}_d = \mathbf{X}_s \cup \mathbf{X}_u$  is first specialised/fine-tuned by the ATTRACT-REPEL specialisation model (or any other post-processing model) to obtain the transformed subspace  $\mathbf{X}'_s$ . The words present (i.e., *seen*) in the input set of linguistic constraints are now assigned different representations in  $\mathbf{X}_s$  (the original distributional vector) and  $\mathbf{X}'_s$  (the specialised vector): they are therefore used as training examples to learn a non-linear cross-space mapping function. This function is then applied to all word vectors  $\mathbf{x}_i \in \mathbf{X}_u$  representing words *unseen* in the constraints to yield a specialised subspace  $\widehat{\mathbf{X}}'_u$ . The final space is  $\mathbf{X}_f = \mathbf{X}'_s \cup \widehat{\mathbf{X}}'_u$ , and it contains transformed representations for *all* words from the initial space  $\mathbf{X}_d$ . **(b)** The actual implementation of the non-linear regression function which maps from  $\mathbf{X}_u$  to  $\widehat{\mathbf{X}}'_u$ : a deep feed-forward fully-connected neural net with non-linearities and  $H$  hidden layers.

least squares may be formulated as follows:

$$f_{\text{MSE}} = \arg \min \|f(\mathbf{X}_s) - \mathbf{X}'_s\|_F^2 \quad (3)$$

where  $\|\cdot\|_F^2$  denotes the squared Frobenius norm. In the most common form  $f(\mathbf{X}_s)$  is simply a linear map/matrix  $\mathbf{W}_f \in \mathbb{R}^{\dim \times \dim}$  (Mikolov et al., 2013a) as follows:  $f(\mathbf{X}) = \mathbf{W}_f \mathbf{X}$ .

After learning  $f$  based on the  $\mathbf{X}_s \rightarrow \mathbf{X}'_s$  transformation, one can simply apply  $f$  to unseen words:  $\widehat{\mathbf{X}}'_u = f(\mathbf{X}_u)$ . This linear mapping model, termed LINEAR-MSE, has an analytical solution (Artetxe et al., 2016), and has been proven to work well with cross-lingual embeddings. However, given that the specialisation model injects hundreds of thousands (or even millions) of linguistic constraints into the distributional space (see later in Sect. 4), we suspect that the assumption of linearity is too limiting and does not fully hold in this particular setup.

Using the same  $L_2$ -penalized least squares objective, we can thus replace the linear map with a *non-linear* function  $f : \mathbb{R}^{\dim} \rightarrow \mathbb{R}^{\dim}$ . The non-linear mapping, illustrated by Fig. 1b, is implemented as a deep feed-forward fully-connected neural network (DFFN) with  $H$  hidden layers and non-linear activations. This variant is called NONLINEAR-MSE.

Another variant objective is the contrastive margin-based ranking loss with negative sampling (MM) similar to the original ATTRACT-REPEL objective, used in other applications in prior work (e.g., for cross-modal mapping) (Weston et al., 2011; Frome et al., 2013; Lazaridou et al., 2015; Kummerfeld et al., 2015). Let  $\widehat{\mathbf{x}}'_i = f(\mathbf{x}_i)$  denote the predicted vector for the word  $x_i \in \mathcal{V}_s$ , and let  $\mathbf{x}'_i$  refer to the “true” vector of  $x_i$  in the specialised space  $\mathbf{X}'_s$  after the AR specialisation procedure. The MM loss is then defined as follows:

$$J_{\text{MM}} = \sum_{i=1}^N \sum_{j \neq i}^k \tau(\delta_{mm} - \cos(\widehat{\mathbf{x}}'_i, \mathbf{x}'_i) + \cos(\widehat{\mathbf{x}}'_i, \mathbf{x}'_j))$$

where  $\cos$  is the cosine similarity measure,  $\delta_{mm}$  is the margin, and  $k$  is the number of negative samples. The objective tries to learn the mapping  $f$  so that each predicted vector  $\widehat{\mathbf{x}}'_i$  is by the specified margin  $\delta_{mm}$  closer to the correct target vector  $\mathbf{x}'_i$  than to any other of  $k$  target vectors  $\mathbf{x}'_j$  serving as negative examples.<sup>3</sup> Function  $f$  can again be either a simple linear map (LINEAR-MM), or implemented as a DFFN (NONLINEAR-MM, see Fig. 1b).

<sup>3</sup>We have also experimented with a simpler *hinge loss* function without negative examples, formulated as  $J =$

## 4 Experimental Setup

### Starting Word Embeddings ( $\mathbf{X}_d = \mathbf{X}_s \cup \mathbf{X}_u$ )

To test the robustness of our approach, we experiment with three well-known, publicly available collections of English word vectors: **1)** Skip-Gram with Negative Sampling (SGNS-BOW2) (Mikolov et al., 2013b) trained on the Polyglot Wikipedia (Al-Rfou et al., 2013) by Levy and Goldberg (2014) using bag-of-words windows of size 2; **2)** GLOVE Common Crawl (Pennington et al., 2014); and **3)** FASTTEXT (Bojanowski et al., 2017), a SGNS variant which builds word vectors as the sum of their constituent character n-gram vectors. All word embeddings are 300-dimensional.<sup>4</sup>

### AR Specialisation and Constraints ( $\mathbf{X}_s \rightarrow \mathbf{X}'_s$ )

We experiment with linguistic constraints used before by (Mrkšić et al., 2017; Vulić et al., 2017a): they extracted monolingual synonymy/ATTRACT pairs from the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013; Pavlick et al., 2015) (640,435 synonymy pairs in total), while their antonymy/REPEL constraints came from BabelNet (Navigli and Ponzetto, 2012) (11,939 pairs).<sup>5</sup>

The coverage of  $\mathcal{V}_d$  vocabulary words in the constraints illustrates well the problem of unseen words with the fine-tuning specialisation models. For instance, the constraints cover only a small subset of the entire vocabulary  $\mathcal{V}_d$  for SGNS-BOW2: 16.6%. They also cover only 15.3% of the top 200K most frequent  $\mathcal{V}_d$  words from FASTTEXT.

### Network Design and Parameters ( $\mathbf{X}_u \rightarrow \widehat{\mathbf{X}}'_u$ )

The non-linear regression function  $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is a DFFN with  $H$  hidden layers, each of dimensionality  $d_1 = d_2 = \dots = d_H = 512$  (see Fig. 1b). Non-linear activations are used in each layer and

$\sum_{i=1}^N \tau \left( \delta_{mm} - \cos(\widehat{\mathbf{x}}'_i, \mathbf{x}'_i) \right)$ . For instance, with  $\delta_{mm} = 1.0$  the idea is to learn a mapping  $f$  that, for each  $x_i$  enforces the predicted vector and the correct target vector to have a maximum cosine similarity. We do not report the results with this variant as, although it outcores the MSE-style objective, it was consistently outperformed by the MM objective.

<sup>4</sup>For further details regarding the architectures and training setup of the used vector collections, we refer the reader to the original papers. Additional experiments with other word vectors, e.g., with CONTEXT2VEC (Melamud et al., 2016a) (which uses bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) for context modeling), and with dependency-word based embeddings (Bansal et al., 2014; Melamud et al., 2016b) lead to similar results and same conclusions.

<sup>5</sup>We have experimented with another set of constraints used in prior work (Zhang et al., 2014; Ono et al., 2015), reaching similar conclusions: these were extracted from WordNet (Fellbaum, 1998) and Roget (Kipfer, 2009), and comprise 1,023,082 synonymy pairs and 380,873 antonymy pairs.

omitted only before the final output layer to enable full-range predictions (see Fig. 1b again).

The choices of non-linear activation and initialisation are guided by recent recommendations from the literature. First, we use *swish* (Ramachandran et al., 2017; Elfwing et al., 2017) as non-linearity, defined as  $swish(x) = x \cdot \text{sigmoid}(\beta x)$ . We fix  $\beta = 1$  as suggested by Ramachandran et al. (2017).<sup>6</sup> Second, we use the HE normal initialisation (He et al., 2015), which is preferred over the XAVIER initialisation (Glorot and Bengio, 2010) for deep models (Mishkin and Matas, 2016; Li et al., 2016), although in our experiments we do not observe a significant difference in performance between the two alternatives. We set  $H = 5$  in all experiments without any fine-tuning; we also analyse the impact of the network depth in Sect. 5.

**Optimisation** For the AR specialisation step, we adopt the original suggested model setup. Hyperparameter values are set to:  $\delta_{att} = 0.6$ ,  $\delta_{rep} = 0.0$ ,  $\lambda_{reg} = 10^{-9}$  (Mrkšić et al., 2017). The models are trained for 5 epochs with Adagrad (Duchi et al., 2011), with batch sizes set to  $b_{att} = b_{rep} = 50$ , again as in the original work.

For training the non-linear mapping with DFFN (Fig. 1b), we use the Adam algorithm (Kingma and Ba, 2015) with default settings. The model is trained for 100 epochs with early stopping on a validation set. We reserve 10% of all available seen data (i.e., the words from  $\mathcal{V}_s$  represented in  $\mathbf{X}_s$  and  $\mathbf{X}'_s$ ) for validation, the rest are used for training. For the MM objective, we set  $\delta_{mm} = 0.6$  and  $k = 25$  in all experiments without any fine-tuning.

## 5 Results and Discussion

### 5.1 Intrinsic Evaluation: Word Similarity

**Evaluation Protocol** The first set of experiments evaluates vector spaces with different specialisation procedures intrinsically on word similarity benchmarks: we use the SimLex-999 dataset (Hill et al., 2015), and SimVerb-3500 (Gerz et al., 2016), a recent verb pair similarity dataset providing similarity ratings for 3,500 verb pairs.<sup>7</sup> Spearman’s  $\rho$

<sup>6</sup>According to Ramachandran et al. (2017), for deep networks *swish* has a slight edge over the family of LU/ReLU-related activations (Maas et al., 2013; He et al., 2015; Klambauer et al., 2017). We also observe a minor (and insignificant) difference in performance in favour of *swish*.

<sup>7</sup>While other gold standards such as WordSim-353 (Finkelstein et al., 2002) or MEN (Bruni et al., 2014) coalesce the notions of true semantic similarity and (more broad) conceptual relatedness, SimLex and SimVerb provide explicit guidelines

	Setup: <i>hold-out</i>						Setup: <i>all</i>					
	GLOVE		SGNS-BOW2		FASTTEXT		GLOVE		SGNS-BOW2		FASTTEXT	
	SL	SV	SL	SV	SL	SV	SL	SV	SL	SV	SL	SV
<b>Distributional: <math>\mathbf{X}_d</math></b>	.408	.286	.414	.275	.383	.255	.408	.286	.414	.275	.383	.255
<b>+AR specialisation: <math>\mathbf{X}'_s</math></b>	.408	.286	.414	.275	.383	.255	.690	.578	.658	.544	.629	.502
<b>++Mapping unseen: <math>\mathbf{X}_f</math></b>												
LINEAR-MSE	.504	.384	.447	.309	.405	.285	.690	.578	.656	.551	.628	.502
NONLINEAR-MSE	.549	.407	.484	.344	.459	.329	.694	.586	.663	.556	.631	.506
LINEAR-MM	.548	.422	.468	.329	.419	.308	.697	.582	.663	.554	.628	.487
NONLINEAR-MM	<b>.603</b>	<b>.480</b>	<b>.531</b>	<b>.391</b>	<b>.471</b>	<b>.349</b>	<b>.705</b>	<b>.600</b>	<b>.667</b>	<b>.562</b>	<b>.638</b>	<b>.507</b>

Table 1: Spearman’s  $\rho$  correlation scores for three word vector collections on two English word similarity datasets, SimLex-999 (SL) and SimVerb-3500 (SV), using different mapping variants, evaluation protocols, and word vector spaces: from the initial distributional space  $\mathbf{X}_d$  to the fully specialised space  $\mathbf{X}_f$ .  $H = 5$ .

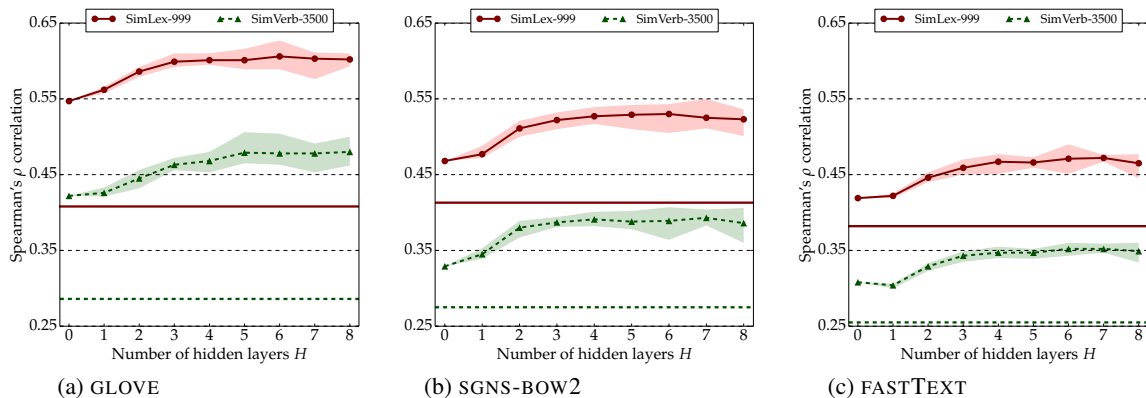


Figure 2: The results of the *hold-out* experiments on SimLex-999 and SimVerb-3500 after applying our non-linear vector space transformation with different depths (hidden layer size  $H$ , see Fig. 1b). The results are presented as averages over 20 runs with the NONLINEAR-MM variant, the shaded regions are spanned by the maximum and minimum scores obtained. Thick horizontal lines refer to Spearman’s rank correlations achieved in the initial space  $\mathbf{X}_d$ .  $H = 0$  denotes the standard *linear* regression model (Mikolov et al., 2013a; Lazaridou et al., 2015) (LINEAR-MM shown since it outperforms LINEAR-MSE).

rank correlation is used as the evaluation metric.

We evaluate word vectors in two settings. First, in a synthetic *hold-out* setting, we remove all linguistic constraints which contain words from the SimLex and SimVerb evaluation data, effectively forcing all SimLex and SimVerb words to be *unseen* by the AR specialisation model. The specialised vectors for these words are estimated by the learned non-linear DFFN mapping model. Second, the *all* setting is a standard “real-life” scenario where some test (SimLex/SimVerb) words do occur in the constraints, while the mapping is learned for the remaining words.

**Results and Analysis** The results with the three word vector collections are provided in Tab. 1. In addition, Fig. 2 plots the influence of the network

to discern between the two, so that related but non-similar words (e.g. *tiger* and *jungle*) have a low rating.

depth  $H$  on the model’s performance.

The results suggest that the mapping of unseen words is universally useful, as the highest correlation scores are obtained with the final fully specialised vector space  $\mathbf{X}_f$  for all three input spaces. The results in the *hold-out* setup are particularly indicative of the improvement achieved by our post-specialisation method. For instance, it achieves a +0.2 correlation gain with GLOVE on both SimLex and SimVerb by specialising vector representations for words present in these datasets *without* seeing a single external constraint which contains any of these words. This suggests that the perturbation of the seen subspace  $\mathbf{X}_s$  by ATTRACT-REPEL contains implicit knowledge that can be propagated to  $\mathbf{X}_u$ , learning better representations for unseen words. We observe small but consistent improvements across the board in the *all* setup. The smaller gains can be explained by the fact that a majority of

SimLex and SimVerb words are present in the external constraints (93.7% and 87.2%, respectively).

The scores also indicate that *both* non-linearity and the chosen objective function contribute to the quality of the learned mapping: largest gains are reported with the NONLINEAR-MM variant which **a)** employs non-linear activations and **b)** replaces the basic mean-squared-error objective with max-margin. The usefulness of the latter has been established in prior work on cross-space mapping learning (Lazaridou et al., 2015). The former indicates that the initial AR transformation is non-linear. It is guided by a large number of constraints; their effect cannot be captured by a simple linear map as in prior work on, e.g., cross-lingual word embeddings (Mikolov et al., 2013a; Ruder et al., 2017).

Finally, the analysis of the network depth  $H$  indicates that going deeper helps only to a certain extent. Adding more layers allows for a richer parametrisation of the network (which is beneficial given the number of linguistic constraints used by AR). This makes the model more expressive, but it seems to saturate with larger  $H$  values.

### Post-Specialisation with Other Post-Processors

We also verify that our post-specialisation approach is not tied to the ATTRACT-REPEL method, and is indeed applicable on top of any post-processing specialisation method. We analyse the impact of post-specialisation in the *hold-out* setting using the original *retrofitting* (*RFit*) model (Faruqui et al., 2015) and *counter-fitting* (*CFit*) (Mrkšić et al., 2016) in lieu of attract-repel. The results on word similarity with the best-performing NONLINEAR-MM variant are summarised in Tab. 2.

The scores again indicate the usefulness of post-specialisation. As expected, the gains are lower

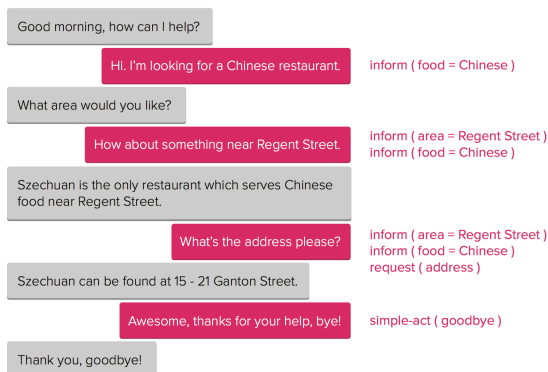


Figure 3: DST labels (user goals given by slot-value pairs) in a multi-turn dialogue (Mrkšić et al., 2015).

	GLOVE		SGNS-BOW2		FASTTEXT	
	SL	SV	SL	SV	SL	SV
$\mathbf{X}_d$	.408	.286	.414	.275	.383	.255
$\mathbf{X}_f$ : <i>RFit</i>	.493	.365	.412	.285	.413	.279
$\mathbf{X}_f$ : <i>CFit</i>	.540	.401	.439	.318	.306	.441

Table 2: Post-specialisation applied to two other post-processing methods. SL: SimLex; SV: SimVerb. *Hold-out* setting. NONLINEAR-MM.

than with ATTRACT-REPEL. *RFit* falls short of *CFit* as by design it can leverage only synonymy (i.e., ATTRACT) external constraints.

### 5.2 Downstream Task I: DST

Next, we evaluate the usefulness of post-specialisation for two downstream tasks – dialogue state tracking and lexical text simplification – in which discerning semantic similarity from other types of semantic relatedness is crucial. We first evaluate the importance of post-specialisation for a downstream language understanding task of *dialogue state tracking* (DST) (Henderson et al., 2014; Williams et al., 2016), adopting the evaluation protocol and data of Mrkšić et al. (2017).

**DST: Model and Evaluation** The DST model is the first component of modern dialogue pipelines (Young, 2010), which captures the users’ goals at each dialogue turn and then updates the dialogue state. Goals are represented as sets of constraints expressed as slot-value pairs (e.g., food=*Chinese*). The set of slots and the set of values for each slot constitute the ontology of a dialogue domain. The probability distribution over the possible states is the system’s estimate of the user’s goals, and it is used by the dialogue manager module to select the subsequent system response (Su et al., 2016). An example in Fig. 3 illustrates the DST pipeline.

For evaluation, we use the Neural Belief Tracker (NBT), a state-of-the-art DST model which was the first to reason purely over pre-trained word vectors (Mrkšić et al., 2017).<sup>8</sup> The NBT uses no hand-crafted semantic lexicons, instead composing word vectors into intermediate utterance and context representations.<sup>9</sup> For full model details, we refer the reader to the original paper. The importance of word vector specialisation for the DST task (e.g., distinguishing between synonyms and antonyms by pulling *northern* and *north* closer in

<sup>8</sup><https://github.com/nmrksic/neural-belief-tracker>

<sup>9</sup>The NBT keeps word vectors *fixed* during training to enable generalisation for words unseen in DST training data.

ENGLISH	<i>hold-out</i>	<i>all</i>
<b>Distributional:</b> $\mathbf{X}_d$	.797	.797
<b>+AR Spec.:</b> $\mathbf{X}'_s \cup \mathbf{X}_u$	.797	.817
<b>++Mapping:</b> $\mathbf{X}_f = \mathbf{X}'_s \cup \widehat{\mathbf{X}}'_u$		
LINEAR-MM	.815	.818
NONLINEAR-MM	<b>.827</b>	<b>.835</b>

Table 3: DST results in two evaluation settings (*hold-out* and *all*) with different GLOVE variants.

the vector space while pushing *north* and *south* away) has been established (Mrkšić et al., 2017).

Again, as in prior work the DST evaluation is based on the Wizard-of-Oz (WOZ) v2.0 dataset (Wen et al., 2017; Mrkšić et al., 2017), comprising 1,200 dialogues split into training (600 dialogues), development (200), and test data (400). In all experiments, we report the standard DST performance measure: *joint goal accuracy*, and report scores as averages over 5 NBT training runs.

**Results and Analysis** We again evaluate word vectors in two settings: **1)** *hold-out*, where linguistic constraints with words appearing in the WOZ data are removed, making all WOZ words unseen by ATTRACT-REPEL; and **2)** *all*. The results for the English DST task with different GLOVE word vector variants are summarised in Tab. 3; similar trends in results are observed with two other word vector collections. The scores maintain conclusions established in the word similarity task. First, semantic specialisation with ATTRACT-REPEL is again beneficial, and discerning between synonyms and antonyms improves DST performance. However, specialising unseen words (the final  $\mathbf{X}_u$  vector space) yields further improvements in both evaluation settings, supporting our claim that the specialisation signal can be propagated to unseen words.

This downstream evaluation again demonstrates the importance of non-linearity, as the peak scores are reported with the NONLINEAR-MM variant. More substantial gains in the *all* setup are observed in the DST task compared to the word similarity task. This stems from a lower coverage of the WOZ data in the AR constraints: 36.3% of all WOZ words are unseen words. Finally, the scores are higher on average in the *all* setup, since this setup uses more external constraints for AR, and consequently uses more training examples to learn the mapping.

**Other Languages** We test the portability of our framework to two other languages for which we have similar evaluation data: German (DE) and

Italian (IT). SimLex-999 has been translated and rescored in the two languages by Leviant and Reichart (2015), and the WOZ data were translated and adapted by Mrkšić et al. (2017). Exactly the same setup is used as in our English experiments, without any additional language-specific fine-tuning. Linguistic constraints were extracted from the same sources: synonyms from the PPDB (135,868 in DE, 362,452 in IT), antonyms from BabelNet (4,124 in DE, and 16,854 in IT). Our starting distributional vector spaces are taken from prior work: IT vectors are from (Dinu et al., 2015), DE vectors are from (Vulić and Korhonen, 2016a). The results are summarised in Tab. 4.

Our post-specialisation approach yields consistent improvements over the initial distributional space and the AR specialisation model in both tasks and for both languages. We do not observe any gain on IT SimLex in the *all* setup since IT constraints have almost complete coverage of all IT SimLex words (99.3%; the coverage is 64.8% in German). As expected, the DST scores in the *all* setup are higher than in the *hold-out* setup due to a larger number of constraints and training examples.

Lower absolute scores for Italian and German compared to the ones reported for English are due to multiple factors, as discussed previously by Mrkšić et al. (2017): **1)** the AR model uses less linguistic constraints for DE and IT; **2)** distributional word vectors are induced from smaller corpora; **3)** linguistic phenomena (e.g., cases and compounding in DE) contribute to data sparsity and also make the DST task more challenging. However, it is important to stress the consistent gains over the vector space specialised by the state-of-the-art ATTRACT-REPEL model across all three test languages. This indicates that the proposed approach is language-agnostic and portable to multiple languages.

### 5.3 Downstream Task II: Lexical Simplification

In our second downstream task, we examine the effects of post-specialisation on lexical simplification (LS) in English. LS aims to substitute complex words (i.e., less commonly used words) with their simpler synonyms in the context. Simplified text must keep the meaning of the original text, which is discerning similarity from relatedness is important (e.g., in “*The automobile was set on fire*” the word “*automobile*” should be replaced with “*car*” or “*vehicle*” but not with “*wheel*” or “*driver*”).



	GERMAN				ITALIAN			
	SimLex (Similarity)		WOZ (DST)		SimLex (Similarity)		WOZ (DST)	
	<i>hold-out</i>	<i>all</i>	<i>hold-out</i>	<i>all</i>	<i>hold-out</i>	<i>all</i>	<i>hold-out</i>	<i>all</i>
<b>Distributional:</b> $X_d$	.267	.267	.487	.487	.363	.363	.618	.618
<b>+AR Spec.:</b> $X'_s \cup X_u$	.267	.422	.487	.535	.363	<b>.616</b>	.618	.634
<b>++Mapping:</b> $X_f$								
LINEAR-MM	.354	.449	.485	.533	.401	<b>.616</b>	.627	.633
NONLINEAR-MM	<b>.367</b>	<b>.466</b>	<b>.496</b>	<b>.538</b>	<b>.428</b>	<b>.616</b>	<b>.637</b>	<b>.647</b>

Table 4: Results on word similarity (Spearman’s  $\rho$ ) and DST (joint goal accuracy) for German and Italian.

Vectors	Specialisation	Acc.	Ch.
GLOVE	<b>Distributional:</b> $X_d$	66.0	<b>94.0</b>
	<b>+AR Spec.:</b> $X'_s \cup X_u$	67.6	87.0
	<b>++Mapping:</b> $X_f$	<b>72.3</b>	87.6
FASTTEXT	<b>Distributional:</b> $X_d$	57.8	84.0
	<b>+AR Spec.:</b> $X'_s \cup X_u$	69.8	<b>89.4</b>
	<b>++Mapping:</b> $X_f$	<b>74.3</b>	88.8
SGNS-BOW2	<b>Distributional:</b> $X_d$	56.0	79.1
	<b>+AR Spec.:</b> $X'_s \cup X_u$	64.4	86.7
	<b>++Mapping:</b> $X_f$	<b>70.9</b>	<b>86.8</b>

Table 5: Lexical simplification performance with post-specialisation applied on three input spaces.

We employ LIGHT-LS (Glavaš and Štajner, 2015), a lexical simplification algorithm that: 1) makes substitutions based on word similarities in a semantic vector space, and 2) can be provided an arbitrary embedding space as input.<sup>10</sup> For a complex word, LIGHT-LS considers the most similar words from the vector space as simplification candidates. Candidates are ranked according to several features, indicating simplicity and fitness for the context (semantic relatedness to the context of the complex word). The substitution is made if the best candidate is simpler than the original word. By providing vector spaces post-specialised for semantic similarity to LIGHT-LS, we expect to more often replace complex words with their true synonyms.

We evaluate LIGHT-LS performance in the *all* setup on the LS benchmark compiled by Horn et al. (2014), who crowdsourced 50 manual simplifications for each complex word. As in prior work, we evaluate performance with the following metrics: 1) *Accuracy* (Acc.) is the number of correct simplifications made (i.e., the system made the simplification and its substitution is found in the list of crowdsourced substitutions), divided by the total number of indicated complex words; 2) *Changed* (Ch.) is the percentage of indicated complex words

<sup>10</sup><https://github.com/codogogo/lightls>

that were replaced by the system (whether or not the replacement was correct).

LS results are summarised in Tab. 5. Post-specialised vector spaces consistently yield 5-6% gain in *Accuracy* compared to respective distributional vectors and embeddings specialised with the state-of-the-art ATTRACT-REPEL model. Similar to DST evaluation, improvements over ATTRACT-REPEL demonstrate the importance of specialising the vectors of the entire vocabulary and not only the vectors of words from the external constraints.

## 6 Conclusion and Future Work

We have presented a novel post-processing model, termed *post-specialisation*, that specialises word vectors for the full vocabulary of the input vector space. Previous post-processing specialisation models fine-tune word vectors only for words occurring in external lexical resources. In this work, we have demonstrated that the specialisation of the subspace of seen words can be leveraged to learn a mapping function which specialises vectors for all other words, unseen in the external resources. Our results across word similarity and downstream language understanding tasks show consistent improvements over the state-of-the-art specialisation method for all three test languages.

In future work, we plan to extend our approach to specialisation for asymmetric relations such as hypernymy or meronymy (Glavaš and Ponzetto, 2017; Nickel and Kiela, 2017; Vulić and Mrkšić, 2018). We will also investigate more sophisticated non-linear functions. The code is available at: <https://github.com/cambridgeltl/post-specialisation/>.

## Acknowledgments

We thank the three anonymous reviewers for their insightful suggestions. This work is supported by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (no 648909).

## References

- Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of CoNLL*, pages 183–192.
- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of EMNLP*, pages 2289–2294.
- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of ACL*, pages 451–462.
- Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of ACL*, pages 809–815.
- Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Knowledge-powered deep learning for word embedding. In *Proceedings of ECML-PKDD*, pages 132–148.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the ACL*, 5:135–146.
- Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.
- Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*, pages 740–750.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.
- Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *CoRR*, abs/1710.04087.
- Ryan Cotterell, Hinrich Schütze, and Jason Eisner. 2016. Morphological smoothing and extrapolation of word embeddings. In *Proceedings of ACL*, pages 1651–1660.
- Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *Proceedings of ICLR (Workshop Papers)*.
- John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.
- Stefan Elfving, Eiji Uchibe, and Kenji Doya. 2017. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *CoRR*, abs/1702.03118.
- Manaal Faruqui. 2016. *Diverse Context for Learning Word Representations*. Ph.D. thesis, Carnegie Mellon University.
- 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 NAACL-HLT*, pages 1606–1615.
- Christiane Fellbaum. 1998. *WordNet*.
- Lev Finkelstein, Evgeniy Gabilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.
- Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc’Aurelio Ranzato, and Tomas Mikolov. 2013. DeViSE: A deep visual-semantic embedding model. In *Proceedings of NIPS*, pages 2121–2129.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of NAACL-HLT*, pages 758–764.
- Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of EMNLP*, pages 2173–2182.
- Goran Glavaš and Simone Paolo Ponzetto. 2017. Dual tensor model for detecting asymmetric lexico-semantic relations. In *Proceedings of EMNLP*, pages 1758–1768.
- Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of ACL*, pages 63–68.
- Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AISTATS*, pages 249–256.
- Zellig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of ICCV*, pages 1026–1034.
- Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The Second Dialog State Tracking Challenge. In *Proceedings of SIGDIAL*, pages 263–272.

- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. [SimLex-999: Evaluating semantic models with \(genuine\) similarity estimation](#). *Computational Linguistics*, 41(4):665–695.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long Short-Term Memory](#). *Neural Computation*, 9(8):1735–1780.
- Colby Horn, Cathryn Manduca, and David Kauchak. 2014. [Learning a lexical simplifier using wikipedia](#). In *Proceedings of the ACL*, pages 458–463.
- Douwe Kiela, Felix Hill, and Stephen Clark. 2015. [Specializing word embeddings for similarity or relatedness](#). In *Proceedings of EMNLP*, pages 2044–2048.
- Joo-Kyung Kim, Gokhan Tur, Asli Celikyilmaz, Bin Cao, and Ye-Yi Wang. 2016. [Intent detection using semantically enriched word embeddings](#). In *Proceedings of SLT*.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *Proceedings of ICLR (Conference Track)*.
- Barbara Ann Kipfer. 2009. *Roget’s 21st Century Thesaurus (3rd Edition)*. Philip Lief Group.
- Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. [Self-normalizing neural networks](#). *CoRR*, abs/1706.02515.
- Jonathan K. Kummerfeld, Taylor Berg-Kirkpatrick, and Dan Klein. 2015. [An empirical analysis of optimization for max-margin NLP](#). In *Proceedings of EMNLP*, pages 273–279.
- Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. [Hubness and pollution: Delving into cross-space mapping for zero-shot learning](#). In *Proceedings of ACL*, pages 270–280.
- Ira Leviant and Roi Reichart. 2015. [Separated by an un-common language: Towards judgment language informed vector space modeling](#). *CoRR*, abs/1508.00106.
- Omer Levy and Yoav Goldberg. 2014. [Dependency-based word embeddings](#). In *Proceedings of ACL*, pages 302–308.
- Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. [Improving distributional similarity with lessons learned from word embeddings](#). *Transactions of the ACL*, 3:211–225.
- Sihan Li, Jiantao Jiao, Yanjun Han, and Tsachy Weissman. 2016. [Demystifying ResNet](#). *CoRR*, abs/1611.01186.
- Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015. [Learning semantic word embeddings based on ordinal knowledge constraints](#). In *Proceedings of ACL*, pages 1501–1511.
- Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. 2013. [Rectifier nonlinearities improve neural network acoustic models](#). In *Proceedings of ICML*.
- Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016a. [Context2vec: Learning generic context embedding with bidirectional LSTM](#). In *Proceedings of CoNLL*, pages 51–61.
- Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016b. [The role of context types and dimensionality in learning word embeddings](#). In *Proceedings of NAACL-HLT*, pages 1030–1040.
- Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. [Exploiting similarities among languages for machine translation](#). *arXiv preprint, CoRR*, abs/1309.4168.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. [Distributed representations of words and phrases and their compositionality](#). In *Proceedings of NIPS*, pages 3111–3119.
- Dmytro Mishkin and Jiri Matas. 2016. [All you need is a good init](#). In *Proceedings of ICLR (Conference Track)*.
- Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. [Neural belief tracker: Data-driven dialogue state tracking](#). In *Proceedings of ACL*, pages 1777–1788.
- Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. [Multi-domain dialog state tracking using recurrent neural networks](#). In *Proceedings of ACL*, pages 794–799.
- Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. [Counter-fitting word vectors to linguistic constraints](#). In *Proceedings of NAACL-HLT*.
- Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. [Semantic specialisation of distributional word vector spaces using monolingual and cross-lingual constraints](#). *Transactions of the ACL*, 5:309–324.
- Vinod Nair and Geoffrey E. Hinton. 2010. [Rectified linear units improve restricted Boltzmann machines](#). In *Proceedings of ICML*, pages 807–814.
- Roberto Navigli and Simone Paolo Ponzetto. 2012. [BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network](#). *Artificial Intelligence*, 193:217–250.

- Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. [Hierarchical embeddings for hypernymy detection and directionality](#). In *Proceedings of EMNLP*, pages 233–243.
- Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. [Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction](#). In *Proceedings of ACL*, pages 454–459.
- Maximilian Nickel and Douwe Kiela. 2017. [Poincaré embeddings for learning hierarchical representations](#). In *Proceedings of NIPS*.
- Masataka Ono, Makoto Miwa, and Yutaka Sasaki. 2015. [Word embedding-based antonym detection using thesauri and distributional information](#). In *Proceedings of NAACL-HLT*, pages 984–989.
- Dominique Osborne, Shashi Narayan, and Shay Cohen. 2016. [Encoding prior knowledge with eigenword embeddings](#). *Transactions of the ACL*, 4:417–430.
- Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. [PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification](#). In *Proceedings of ACL*, pages 425–430.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global vectors for word representation](#). In *Proceedings of EMNLP*, pages 1532–1543.
- Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2017. [Searching for activation functions](#). *CoRR*, abs/1710.05941.
- Sascha Rothe and Hinrich Schütze. 2015. [AutoExtend: Extending word embeddings to embeddings for synsets and lexemes](#). In *Proceedings of ACL*, pages 1793–1803.
- Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2017. [A survey of cross-lingual embedding models](#). *CoRR*, abs/1706.04902.
- Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. [Symmetric pattern based word embeddings for improved word similarity prediction](#). In *Proceedings of CoNLL*, pages 258–267.
- Pei-Hao Su, Milica Gašić, Nikola Mrkšić, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. [Continuously learning neural dialogue management](#). In *arXiv preprint: 1606.02689*.
- Ivan Vulić and Anna Korhonen. 2016a. [Is "universal syntax" universally useful for learning distributed word representations?](#) In *Proceedings of ACL*, pages 518–524.
- Ivan Vulić and Anna Korhonen. 2016b. [On the role of seed lexicons in learning bilingual word embeddings](#). In *Proceedings of ACL*, pages 247–257.
- Ivan Vulić and Nikola Mrkšić. 2018. [Specialising word vectors for lexical entailment](#). In *Proceedings of NAACL-HLT*.
- Ivan Vulić, Nikola Mrkšić, and Anna Korhonen. 2017a. [Cross-lingual induction and transfer of verb classes based on word vector space specialisation](#). In *Proceedings of EMNLP*, pages 2536–2548.
- Ivan Vulić, Nikola Mrkšić, Roi Reichart, Diarmuid Ó Séaghdha, Steve Young, and Anna Korhonen. 2017b. [Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules](#). In *Proceedings of ACL*, pages 56–68.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. [A network-based end-to-end trainable task-oriented dialogue system](#). In *Proceedings of EACL*.
- Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. [WSABIE: Scaling up to large vocabulary image annotation](#). In *Proceedings of IJCAI*, pages 2764–2770.
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. [From paraphrase database to compositional paraphrase model and back](#). *Transactions of the ACL*, 3:345–358.
- Jason D. Williams, Antoine Raux, and Matthew Henderson. 2016. [The Dialog State Tracking Challenge series: A review](#). *Dialogue & Discourse*, 7(3):4–33.
- Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. [RC-NET: A general framework for incorporating knowledge into word representations](#). In *Proceedings of CIKM*, pages 1219–1228.
- Steve Young. 2010. [Cognitive User Interfaces](#). *IEEE Signal Processing Magazine*.
- Mo Yu and Mark Dredze. 2014. [Improving lexical embeddings with semantic knowledge](#). In *Proceedings of ACL*, pages 545–550.
- Jingwei Zhang, Jeremy Salwen, Michael Glass, and Alfio Gliozzo. 2014. [Word semantic representations using bayesian probabilistic tensor factorization](#). In *Proceedings of EMNLP*, pages 1522–1531.