Explicit Retrofitting of Distributional Word Vectors

Goran Glavaš
Data and Web Science Group
University of Mannheim
B6, 29, DE-68161 Mannheim
goran@informatik.uni-mannheim.de

Ivan Vulić
Language Technology Lab
University of Cambridge
9 West Road, Cambridge CB3 9DA
iv250@cam.ac.uk

Abstract

Semantic specialization of distributional word vectors, referred to as *retrofitting*, is a process of fine-tuning word vectors using external lexical knowledge in order to better embed some semantic relation. Existing retrofitting models integrate linguistic constraints directly into learning objectives and, consequently, specialize only the vectors of words from the constraints. In this work, in contrast, we transform external lexico-semantic relations into training examples which we use to learn an *explicit retrofitting model (ER)*. The ER model allows us to learn a global specialization function and specialize the vectors of words unobserved in the training data as well. We report large gains over original distributional vector spaces in (1) intrinsic word similarity evaluation and on (2) two downstream tasks – *lexical simplification* and *dialog state tracking*. Finally, we also successfully specialize vector spaces of new languages (i.e., unseen in the training data) by coupling ER with shared multilingual distributional vector spaces.

1 Introduction

Algebraic modeling of word vector spaces is one of the core research areas in modern Natural Language Processing (NLP) and its usefulness has been shown across a wide variety of NLP tasks (Collobert et al., 2011; Chen and Manning, 2014; Melamud et al., 2016). Commonly employed *distributional* models for word vector induction are based on the distributional hypothesis (Harris, 1954), i.e., they rely on word co-occurrences obtained from large text corpora (Mikolov et al., 2013b; Pennington et al., 2014; Levy and Goldberg, 2014a; Levy et al., 2015; Bojanowski et al., 2017).

The dependence on purely distributional knowledge results in a well-known tendency of fusing semantic similarity with other types of semantic relatedness (Hill et al., 2015; Schwartz et al., 2015) in the induced vector spaces. Consequently, the similarity between distributional vectors indicates just an abstract semantic association and not a precise semantic relation (Yih et al., 2012; Mohammad et al., 2013). For example, it is difficult to discern synonyms from antonyms in distributional spaces. This property has a particularly negative effect on NLP applications like text simplification and statistical dialog modeling, in which discerning semantic similarity from other types of semantic relatedness is pivotal to the system performance (Glavaš and Štajner, 2015; Faruqui et al., 2015; Mrkšić et al., 2016; Kim et al., 2016b).

A standard solution is to move beyond purely unsupervised learning of word representations, in a process referred to as *word vector space specialization* or *retrofitting*. Specialization models leverage external lexical knowledge from lexical resources, such as WordNet (Fellbaum, 1998), the Paraphrase Database (Ganitkevitch et al., 2013), or BabelNet (Navigli and Ponzetto, 2012), to *specialize* distributional spaces for a particular lexical relation, e.g., synonymy (Faruqui et al., 2015; Mrkšić et al., 2017) or hypernymy (Glavaš and Ponzetto, 2017). External constraints are commonly pairs of words between which a particular relation holds.

Existing specialization methods exploit the external linguistic constraints in two prominent ways: (1) *joint specialization* models modify the learning objective of the original distributional model by integrating the constraints into it (Yu and Dredze, 2014; Kiela et al., 2015; Nguyen et al., 2016, *inter alia*); (2) *post-processing* models fine-tune distributional vectors retroactively after training to satisfy the external constraints (Faruqui et al., 2015;

Mrkšić et al., 2017, *inter alia*). The latter, in general, outperform the former (Mrkšić et al., 2016). Retrofitting models can be applied to arbitrary distributional spaces but they suffer from a major limitation – they *locally* update only vectors of words present in the external constraints, whereas vectors of all other (unseen) words remain intact. In contrast, joint specialization models propagate the external signal to all words via the joint objective.

In this paper, we propose a new approach for specializing word vectors that unifies the strengths of both prior strategies, while mitigating their limitations. Same as retrofitting models, our novel framework, termed *explicit retrofitting (ER)*, is applicable to *arbitrary* distributional spaces. At the same time, the method learns an *explicit global specialization* function that can specialize vectors for *all* vocabulary words, similar as in joint models. Yet, unlike the joint models, ER does not require expensive re-training on large text corpora, but is directly applied on top of any pre-trained vector space. The key idea of ER is to directly learn a specialization function in a *supervised* setting, using lexical constraints as *training instances*. In other words, our model, implemented as a deep feed-forward neural architecture, learns a (non-linear) function which "translates" word vectors from the distributional space into the specialized space.

We show that the proposed ER approach yields considerable gains over distributional spaces in word similarity evaluation on standard benchmarks (Hill et al., 2015; Gerz et al., 2016), as well as in two downstream tasks – lexical simplification and dialog state tracking. Furthermore, we show that, by coupling the ER model with shared multilingual embedding spaces (Mikolov et al., 2013a; Smith et al., 2017), we can also specialize distributional spaces for languages unseen in the training data in a zero-shot language transfer setup. In other words, we show that an explicit retrofitting model trained with external constraints from one language can be successfully used to specialize the distributional space of another language.

2 Related Work

The importance of vector space specialization for downstream tasks has been observed, *inter alia*, for dialog state tracking (Mrkšić et al., 2017; Vulić et al., 2017b), spoken language understanding (Kim et al., 2016b,a), judging lexical entailment (Nguyen et al., 2017; Glavaš and Ponzetto, 2017; Vulić and Mrkšić, 2017), lexical contrast modeling (Nguyen et al., 2016), and cross-lingual transfer of lexical resources (Vulić et al., 2017a). A common goal pertaining to all retrofitting models is to pull the vectors of similar words (e.g., synonyms) closer together, while some models also push the vectors of dissimilar words (e.g., antonyms) further apart. The specialization methods fall into two categories: (1) joint specialization methods, and (2) post-processing (i.e., retrofitting) methods. Methods from both categories make use of similar lexical resources – they typically leverage WordNet (Fellbaum, 1998), FrameNet (Baker et al., 1998), the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013; Pavlick et al., 2015), morphological lexicons (Cotterell et al., 2016), or simple handcrafted linguistic rules (Vulić et al., 2017b). In what follows, we discuss the two model categories.

Joint Specialization Models. These models integrate external constraints into the distributional training procedure of general word embedding algorithms such as CBOW, Skip-Gram (Mikolov et al., 2013b), or Canonical Correlation Analysis (Dhillon et al., 2015). They modify the prior or the regularization of the original objective (Yu and Dredze, 2014; Xu et al., 2014; Bian et al., 2014; Kiela et al., 2015) or integrate the constraints directly into the, e.g., an SGNS- or CBOW-style objective (Liu et al., 2015; Ono et al., 2015; Bollegala et al., 2016; Osborne et al., 2016; Nguyen et al., 2016, 2017). Besides generally displaying lower performance compared to retrofitting methods (Mrkšić et al., 2016), these models are also tied to the distributional objective and any change of the underlying distributional model induces a change of the entire joint model. This makes them less versatile than the retrofitting methods.

Post-Processing Models. Models from the popularly termed *retrofitting* family inject lexical knowledge from external resources into arbitrary pre-trained word vectors (Faruqui et al., 2015; Jauhar et al., 2015; Rothe and Schütze, 2015; Wieting et al., 2015; Nguyen et al., 2016; Mrkšić et al., 2016). These models fine-tune the vectors of words present in the linguistic constraints to reflect the ground-truth lexical knowledge. While the large majority of specialization models from both classes operate only with similarity constraints, a line of recent work (Mrkšić et al., 2016; Mrkšić et al., 2017; Vulić et al., 2017b) demonstrates that knowledge about both similar and dissimilar words leads to

improved performance in downstream tasks. The main shortcoming of the existing retrofitting models is their inability to specialize vectors of words unseen in external lexical resources.

Our explicit retrofitting framework brings together desirable properties of both model classes: (1) unlike joint models, it does not require adaptation to the underlying distributional model and expensive re-training, i.e., it is applicable to any pre-trained distributional space; (2) it allows for easy integration of both similarity and dissimilarity constraints into the specialization process; and (3) unlike post-processors, it specializes the full vocabulary of the original distributional space and not only vectors of words from external constraints.

3 Explicit Retrofitting

Our explicit retrofitting (ER) approach, illustrated by Figure 1a, consists of two major components: (1) an algorithm for preparing training instances from external lexical constraints, and (2) a supervised specialization model, based on a deep feed-forward neural network. This network, shown in Figure 1b learns a non-linear global specialization function from the training instances.

3.1 From Constraints to Training Instances

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, $\mathbf{x}_i \in \mathbb{R}^d$ be the d-dimensional distributional vector space that we want to specialize (with $V = \{w_i\}_{i=1}^{N}$ referring to the associated vocabulary) and let $\mathbf{X}' = \{\mathbf{x}'_i\}_{i=1}^{N}$ be the corresponding specialized vector space that we seek to obtain through explicit retrofitting. Let $\mathbf{C} = \{(w_i, w_j, r)_l\}_{l=1}^{L}$ be the set of L linguistic constraints from an external lexical resource, each consisting of a pair of vocabulary words w_i and w_j and a semantic relation r that holds between them. The most recent state-of-the-art retrofitting work (Mrkšić et al., 2017; Vulić et al., 2017b) suggests that using both similarity and dissimilarity constraints leads to better performance compared to using only similarity constraints. Therefore, we use synonymy and antonymy relations from external resources, i.e., $r_l \in \{ant, syn\}$. Let g be the function measuring the distance between words w_i and w_j based on their vector representations. The algorithm for preparing training instances from constraints is guided by the following assumptions:

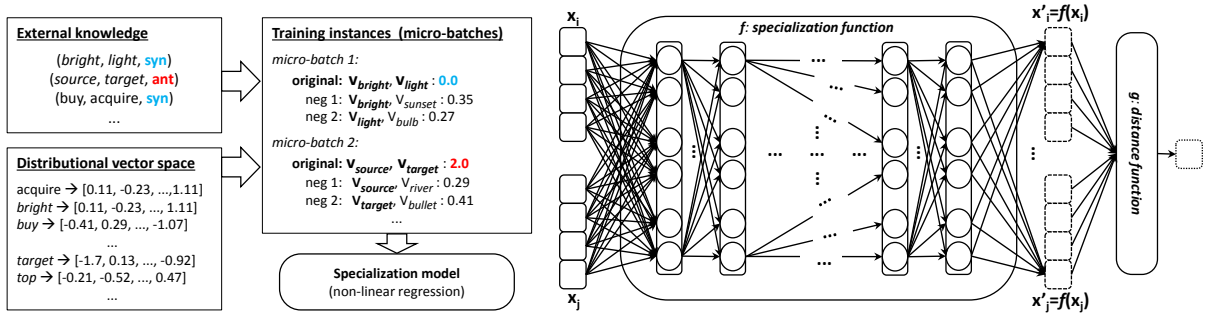
1. All synonymy pairs (w_i, w_j, syn) should have a minimal possible distance score in the specialized space, i.e., $g(\mathbf{x}'_i, \mathbf{x}'_j) = g_{min}$;[1]
2. All antonymy pairs (w_i, w_j, ant) should have a maximal distance in the specialized space, i.e., $g(\mathbf{x}'_i, \mathbf{x}'_j) = g_{max}$;[2]
3. The distances $g(\mathbf{x}'_i, \mathbf{x}'_k)$ in the specialized space between some word w_i and all other words w_k that are not synonyms or antonyms of w_i should be in the interval (g_{min}, g_{max}).

Our goal is to discern semantic similarity from semantic relatedness by comparing, in the specialized space, the distances between word pairs $(w_i, w_j, r) \in \mathbf{C}$ with distances that words w_i and w_j from those pairs have with other vocabulary words w_m. It is intuitive to enforce that the synonyms are as close as possible and antonyms as far as possible. However, we do not know what the distances between non-synonymous and non-antonymous words $g(\mathbf{x}'_i, \mathbf{x}_m)$ in the specialized space should look like. This is why, for all other words, similar to (Faruqui et al., 2016; Mrkšić et al., 2017), we assume that the distances in the specialized space for all word pairs not found in C should stay the same as in the distributional space: $g(\mathbf{x}'_i, \mathbf{x}'_m) = g(\mathbf{x}_i, \mathbf{x}_m)$. This way we preserve the useful semantic content available in the original distributional space.

In downstream tasks most errors stem from vectors of semantically related words (e.g., *car – driver*) being as similar as vectors of semantically similar words (e.g., *car – automobile*). To anticipate this, we compare the distances of pairs $(w_i, w_j, r) \in \mathbf{C}$ with the distances for pairs (w_i, w_m) and (w_j, w_n), where w_m and w_n are *negative examples*: the vocabulary words that are *most similar* to w_i and w_j, respectively, in the original distributional space $\mathbf{X}$. Concretely, for each constraint $(w_i, w_j, r) \in \mathbf{C}$ we retrieve (1) K vocabulary words $\{w_m^k\}_{k=1}^{K}$ that are closest in the input distributional space (according to the distance function g) to the word w_i and (2) K vocabulary words $\{w_n^k\}_{k=1}^{K}$ that are closest to the word w_j. We then create, for each constraint $(w_i, w_j, r) \in \mathbf{C}$, a corresponding set M (termed *micro-batch*) of $2K + 1$ embedding pairs coupled with a corresponding distance in the input distributional space:

[1] The minimal distance value is $g_{min} = 0$ for, e.g., cosine distance or Euclidean distance.

[2] While some distance functions do have a theoretical maximum (e.g., $g_{max} = 2$ for cosine distance), others (e.g., Euclidean distance) may be theoretically unbounded. For unbounded distance measures, we propose using the maximal distance between any two words from the vocabulary as g_{max}.



(a) Illustration of the explicit retrofitting approach

(b) Supervised specialization model

Figure 1: **(a)** High-level illustration of the explicit retrofitting approach: lexical constraints, i.e., pairs of synonyms and antonyms, are transformed into respective micro-batches, which are then used to train the supervised specialization model. **(b)** The low-level implementation of the specialization model, combining the non-linear embedding specialization function f, defined as the deep fully-connected feed-forward network, with the distance metric g, measuring the distance between word vectors after their specialization.

$$M(w_i, w_j, r) = \{(\mathbf{x}_i, \mathbf{x}_j, g_r)\} \cup \{(\mathbf{x}_i, \mathbf{x}^k_m, g(\mathbf{x}_i, \mathbf{x}^k_m))\}_{k=1}^{K} \cup \{(\mathbf{x}_j, \mathbf{x}^k_n, g(\mathbf{x}_j, \mathbf{x}^k_n))\}_{k=1}^{K} \quad (1)$$

with $g_r = g_{min}$ if $r = syn$; $g_r = g_{max}$ if $r = ant$.

3.2 Non-Linear Specialization Function

Our retrofitting framework learns a global *explicit specialization function* which, when applied on a distributional vector space, transforms it into a space that better captures semantic similarity, i.e., discerns similarity from all other types of semantic relatedness. We seek the optimal parameters θ of the parametrized function $f(\mathbf{x}; \theta) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ (where d is the dimensionality of the input space). The specialized embedding $\mathbf{x'}_i$ of the word w_i is then obtained as $\mathbf{x'}_i = f(\mathbf{x}_i; \theta)$. The specialized space $\mathbf{X'}$ is obtained by transforming distributional vectors of *all* vocabulary words, $\mathbf{X'} = f(\mathbf{X}; \theta)$.

We define the specialization function f to be a multi-layer fully-connected feed-forward network with H hidden layers and non-linear activations ϕ. The illustration of this network is given in Figure 1b. The i-th hidden layer is defined with a weight matrix $\mathbf{W}^i$ and a bias vector $\mathbf{b}^i$:

$$h^i(\mathbf{x}; \theta_i) = \phi\left(h^{i-1}(\mathbf{x}; \theta_{i-1})\mathbf{W}^i + \mathbf{b}^i\right) \quad (2)$$

where θ_i is the subset of network's parameters up to the i-th layer. Note that in this notation, $\mathbf{x} = h^0(\mathbf{x}; \emptyset)$ and $\mathbf{x'} = f(\mathbf{x}, \theta) = h^H(\mathbf{x}; \theta)$. Let d_h be the size of the hidden layers. The network's parameters are then as follows: $\mathbf{W}^1 \in \mathbb{R}^{d \times d_h}$; $\mathbf{W}^i \in \mathbb{R}^{d_h \times d_h}$, $i \in \{2, \ldots, H-1\}$; $\mathbf{W}^H \in \mathbb{R}^{d_h \times d}$; $\mathbf{b}^i \in \mathbb{R}^{d_h}$, $i \in \{1, \ldots, H-1\}$; $\mathbf{b}^H \in \mathbb{R}^d$.

3.3 Optimization Objectives

We feed the micro-batches consisting of $2K + 1$ training instances to the specialization model (see Section 3.1). Each training instance consists of a pair of distributional (i.e., unspecialized) embedding vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ and a score g denoting the desired distance between the specialized vectors $\mathbf{x'}_i$ and $\mathbf{x'}_j$ of corresponding words w_i and w_j.

Mean Square Distance Objective (ER-MSD). Let our training batch consist of N training instances, $\{(\mathbf{x}^i_1, \mathbf{x}^i_2, g^i)\}_{i=1}^{N}$. The simplest objective function is then the difference between the desired and obtained distances of specialized vectors:

$$J_{MSD} = \sum_{i=1}^{N}\left(g(f(\mathbf{x}^i_1), f(\mathbf{x}^i_2)) - g^i\right)^2 \quad (3)$$

By minimizing the MSD objective we simply force the specialization model to produce a specialized embedding space $\mathbf{X'}$ in which distances between all synonyms amount to g_{min}, distances between all antonyms amount to g_{max} and distances between all other word pairs remain the same as in the original space. The MSD objective does not leverage negative examples: it only indirectly enforces that synonym (or antonym) pairs (w_i, w_j) have smaller (or larger) distances than corresponding non-constraint word pairs (w_i, w_k) and (w_j, w_k).

Contrastive Objective (ER-CNT). An alternative to MSD is to *directly* contrast the distances of constraint pairs (i.e., antonyms and synonyms)

with the distances of their corresponding negative examples, i.e., the pairs from their respective micro-batch (cf. Eq. (1) in Section 3.1). Such an objective should directly enforce that the similarity scores for synonyms (antonyms) (w_i, w_j) are larger (or smaller, for antonyms) than for pairs (w_i, w_k) and (w_j, w_k) involving the same words w_i and w_j, respectively. Let S and A be the sets of micro-batches created from synonymy and antonymy constraints. Let $M_s = \{(\mathbf{x}_1^i, \mathbf{x}_2^i, g^i)\}_{i=1}^{2K+1}$ be one micro-batch created from one synonymy constraint and let M_a be the analogous micro-batch created from one antonymy constraint. Let us then assume that the first triple (i.e., for $i = 1$) in every micro-batch corresponds to the constraint pair and the remaining $2K$ triples (i.e., for $i \in \{2, \ldots, 2K+1\}$) to respective non-constraint word pairs. We then define the contrastive objective as follows:

$$J_{CNT} = \sum_{M_s \in S} \sum_{i=2}^{2K+1} \left((g^i - g_{min}) - (g'^i - g'^1) \right)^2 + \sum_{M_a \in A} \sum_{i=2}^{2K+1} \left((g_{max} - g^i) - (g'^1 - g'^i) \right)^2$$

where g' is a short-hand notation for the distance between vectors in the specialized space, i.e., $g'(\mathbf{x}_1, \mathbf{x}_2) = g(\mathbf{x}'_1, \mathbf{x}'_2) = g(f(\mathbf{x}_1), f(\mathbf{x}_2))$.

Topological Regularization. Because the distributional space $\mathbf{X}$ already contains useful semantic information, we want our specialized space $\mathbf{X}'$ to move similar words closer together and dissimilar words further apart, but without disrupting the overall topology of $\mathbf{X}$. To this end, we define an additional regularization objective that measures the distance between the original vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ and their specialized counterparts $\mathbf{x}'_1 = f(\mathbf{x}_1)$ and $\mathbf{x}'_2 = f(\mathbf{x}_2)$, for all examples in the training set:

$$J_{REG} = \sum_{i=1}^{N} g(\mathbf{x}_1^i, f(\mathbf{x}_1^i)) + g(\mathbf{x}_2^i, f(\mathbf{x}_2^i)) \quad (4)$$

We minimize the final objective function $J' = J + \lambda J_{REG}$. J is either J_{MSD} or J_{CNT} and λ is the regularization factor which determines how strictly we retain the topology of the original space.

4 Experimental Setup

Distributional Vectors. In order to estimate the robustness of the proposed explicit retrofitting procedure, we experiment with three different publicly available and widely used collections of pre-trained distributional vectors for English: (1) SGNS-W2 – vectors trained on the Wikipedia dump from the Polyglot project (Al-Rfou et al., 2013) using the Skip-Gram algorithm with Negative Sampling (SGNS) (Mikolov et al., 2013b) by Levy and Goldberg (2014b), using the context windows of size 2; (2) GLOVE-CC – vectors trained with the GloVe (Pennington et al., 2014) model on the Common Crawl; and (3) FASTTEXT – vectors trained on Wikipedia with a variant of SGNS that builds word vectors by summing the vectors of their constituent character n-grams (Bojanowski et al., 2017).

Linguistic Constraints. We experiment with the sets of linguistic constraints used in prior work (Zhang et al., 2014; Ono et al., 2015). These constraints, extracted from WordNet (Fellbaum, 1998) and Roget's Thesaurus (Kipfer, 2009), comprise a total of 1,023,082 synonymy word pairs and 380,873 antonymy word pairs.

Although this seems like a large number of linguistic constraints, there is only 57,320 unique words in all synonymy and antonymy constraints combined, and not all of these words are found in the dictionary of the pre-trained distributional vector space. For example, only 15.3% of the words from constraints are found in the whole vocabulary of SGNS-W2 embeddings. Similarly, we find only 13.3% and 14.6% constraint words among the 200K most frequent words from the GLOVE-CC and FASTTEXT vocabularies, respectively. This low coverage emphasizes the core limitation of current retrofitting methods, being able to specialize only the vectors of words seen in the external constraints, and the need for our global ER method which can specialize all word vectors from the distributional space.

ER Model Configuration. In all experiments, we set the distance function g to cosine distance: $g(\mathbf{x}_1, \mathbf{x}_2) = 1 - (\mathbf{x}_1 \cdot \mathbf{x}_2 / (\|\mathbf{x}_1\| \|\mathbf{x}_2\|))$ and use the hyperbolic tangent as activation, $\phi = \tanh$. For each constraint (w_i, w_j), we create $K = 4$ corresponding negative examples for both w_i and w_j, resulting in micro-batches with $2K + 1 = 9$ training instances.[3] We separate 10% of the created micro-batches as the validation set. We then tune the hyper-parameter values, the number of hidden layers $H = 5$ and their size $d_h = 1000$, and the

[3] For $K < 4$ we observed significant performance drop. Setting $K > 4$ resulted in negligible performance gains but significantly increased the model training time.

topological regularization factor $\lambda = 0.3$ by minimizing the model's objective J' on the validation set. We train the model in mini-batches, each containing $N_b = 100$ constraints (i.e., 900 training instances, see above), using the Adam optimizer (Kingma and Ba, 2015) with initial learning rate set to 10^{-4}. We use the loss on the validation set as the early stopping criteria.

5 Results and Discussion

5.1 Word Similarity

Evaluation Setup. We first evaluate the quality of the explicitly retrofitted embedding spaces intrinsically, on two word similarity benchmarks: SimLex-999 dataset (Hill et al., 2015) and SimVerb-3500 (Gerz et al., 2016), a recent dataset containing human similarity ratings for 3,500 verb pairs.[4] We use Spearman's ρ rank correlation between gold and predicted word pair scores as the evaluation metric. We evaluate the specialized embedding spaces in two settings. In the first setting, termed *lexically disjoint*, we remove from our training set all linguistic constraints that contain any of the words found in SimLex or SimVerb. This way, we effectively evaluate the model's ability to generalize the specialization function to *unseen* words. In the second setting (*lexical overlap*) we retain the constraints containing SimLex or SimVerb words in the training set. For comparison, we also report performance of the state-of-the-art local retrofitting model ATTRACT-REPEL (Mrkšić et al., 2017), which is able to specialize only the words from the linguistic constraints.

Results. The results with our ER model applied to three distributional spaces are shown in Table 1. The scores suggest that the proposed ER model is universally useful and robust. The ER-specialized spaces outperform original distributional spaces across the board, for both objective functions. The results in the *lexically disjoint* setting are especially indicative of the improvements achieved by the ER. For example, we achieve a correlation gain of 18% for the GLOVE-CC vectors on SimLex using a specialization function learned *without* seeing a single constraint with any SimLex word.

In the *lexical overlap* setting, we observe substantial gains only for GLOVE-CC. The modest gains in this setting with FASTTEXT and SGNS-W2 in fact strengthen the impression that the ER model learns a *general* specialization function, i.e., it does not "overfit" to words from linguistic constraints. The ER model with the contrastive objective (ER-CNT) yields better performance on average than the one using the simpler square distance objective (ER-MSD). This is expected, given that the contrastive objective enforces the model to distinguish pairs of semantically (dis)similar words from pairs of semantically related words.

Finally, the post-processing ATTRACT-REPEL model based on local vector updates seems to substantially outperform the ER method in this task. The gap is especially visible for FASTTEXT and SGNS-W2 vectors. However, since ATTRACT-REPEL specializes only words seen in linguistic constraints,[5] its performance crucially depends on the coverage of test set words in the constraints. ATTRACT-REPEL excels on the intrinsic evaluation as the constraints cover 99.2% of SimLex words and 99.9% of SimVerb words. However, its usefulness is less pronounced in real-life downstream scenarios in which such high coverage cannot be guaranteed, as demonstrated in Section 5.3.

Analysis. We examine in more detail the performance of the ER model with respect to (1) the type of constraints used for training the model: synonyms and antonyms, only synonyms, or only antonyms and (2) the extent to which we retain the topology of the original distributional space (i.e., with respect to the value of the topological regularization factor λ). All reported results were obtained by specializing the GLOVE-CC distributional space in the *lexically disjoint* setting (i.e., employed constraints did not contain any of the SimLex or SimVerb words).

In Table 2 we show the specialization performance of the ER-CNT models ($H = 5$, $\lambda = 0.3$), using different types of constraints on SimLex-999 (SL) and SimVerb-3500 (SV). We compare the standard model, which exploits both synonym and antonym pairs for creating training instances, with the models employing only synonym and only antonym constraints, respectively. Clearly, we obtain the best specialization when combining synonyms and antonyms. Note, however, that using

[4] Other word similarity datasets such as MEN (Bruni et al., 2014) or WordSim-353 (Finkelstein et al., 2002) conflate the concepts of true semantic similarity and semantic relatedness in a broader sense. In contrast, SimLex and SimVerb explicitly discern between the two, with pairs of semantically related but not similar words (e.g. *car* and *wheel*) having low ratings.

[5] This is why ATTRACT-REPEL cannot be applied in the *lexically disjoint* setting: the scores simply stay the same.

	Setting: *lexically disjoint*						Setting: *lexical overlap*					
	GLOVE-CC		FASTTEXT		SGNS-W2		GLOVE-CC		FASTTEXT		SGNS-W2	
	SL	SV	SL	SV	SL	SV	SL	SV	SL	SV	SL	SV
Distributional ($\mathbf{X}$)	.407	.280	.383	.247	.414	.272	.407	.280	.383	.247	.414	.272
ATTRACT-REPEL	.407	.280	.383	.247	.414	.272	**.690**	**.578**	**.629**	**.502**	**.658**	**.544**
ER-Specialized ($\mathbf{X}' = f(\mathbf{X})$)												
ER-MSD	.483	.345	.429	**.275**	**.445**	.302	.500	.358	**.445**	.284	**.469**	.323
ER-CNT	**.582**	**.439**	**.433**	.272	.435	**.329**	**.623**	**.519**	.419	**.335**	.449	**.355**

Table 1: Spearman's ρ correlation scores for three standard English distributional vectors spaces on English SimLex-999 (SL) and SimVerb-3500 (SV), using explicit retrofitting models with two different objective functions (ER-MSD and ER-CNT, cf. Section 3.3).

Constraints (ER-CNT model)	SL	SV
Synonyms only	.465	.339
Antonyms only	.451	.317
Synonyms + Antonyms	**.582**	**.439**

Table 2: Performance (ρ) on SL and SV for ER-CNT models trained with different constraints.

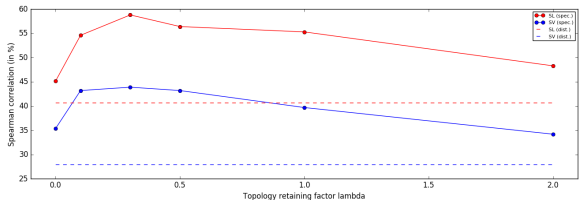


Figure 2: Specialization performance on SimLex-999 (blue line) and SimVerb-3500 (red line) for ER models with different topology regularization factors λ. Dashed lines indicate performance levels of the distributional (i.e., unspecialized) space.

only synonyms or only antonyms also improves over the original distributional space.

Next, in Figure 2 we depict the specialization performance (on SimLex and SimVerb) of the ER models with different values of the topology regularization factor λ (H fixed to 5). The best performance for is obtained for $\lambda = 0.3$. Smaller lambda values overly distort the original distributional space, whereas larger lambda values dampen the specialization effects of linguistic constraints.

5.2 Language Transfer

Readily available large collections of synonymy and antonymy word pairs do not exist for many languages. This is why we also investigate *zero-shot specialization*: we test if it is possible, with the help of cross-lingual word embeddings, to transfer the specialization knowledge learned from English constraints to languages without any training data.

Evaluation Setup. We use the mapping model of Smith et al. (2017) to induce a multilingual vector space[6] containing word vectors of three other languages – German, Italian, and Croatian – along with the English vectors.[7] Concretely, we map the Italian CBOW vectors (Dinu et al., 2015), German FastText vectors trained on German Wikipedia (Bojanowski et al., 2017), and Croatian Skip-Gram vectors trained on HrWaC corpus (Ljubešić and Erjavec, 2011) to the GLOVE-CC English space. We create the translation pairs needed to learn the projections by automatically translating 4,000 most frequent English words to all three other languages with Google Translate. We then employ the ER model trained to specialize the GLOVE-CC space using the full set of English constraints, to specialize the distributional spaces of other languages. We evaluate the quality of the specialized spaces on the respective SimLex-999 dataset for each language (Leviant and Reichart, 2015; Mrkšić et al., 2017).

Model	German	Italian	Croatian
Distributional ($\mathbf{X}$)	.407	.360	.249
ER-Specialized ($\mathbf{X}'$)			
ER-MSD	.415	.406	.287
ER-CNT	**.533**	**.448**	**.315**

Table 3: Spearman's ρ correlation scores for German, Italian, and Croatian embeddings in the transfer setup: the vectors are specialized using the models trained on English constraints and evaluated on respective language-specific SimLex-999 variants.

Results. The results are provided in Table 3. They indicate that the ER models can substantially improve (e.g., by 13% for German vector space) over distributional spaces also in the language transfer setup without seeing a single constraint in the target language. These transfer results hold promise to support vector space specialization

[6]This model was chosen for its ease of use, readily available implementation, and strong comparative results (see (Ruder et al., 2017)). For more details we refer the reader to the original paper and the survey.

[7]The choice of languages was determined by the availability of the language-specific SimLex-999 variants.

even for resource-lean languages. The more sophisticated contrastive ER-CNT model variant again outperforms the simpler ER-MSD variant, and it does so for all three languages, which is consistent with the findings from the monolingual English experiments (see Table 1).

5.3 Downstream Tasks

We now evaluate the impact of our global ER method on two downstream tasks in which differentiating semantic similarity from semantic relatedness is particularly important: lexical text simplification (LS) and dialog state tracking (DST).

5.3.1 Lexical Text Simplification

Lexical simplification aims to replace complex words – used less frequently and known to fewer speakers – with their simpler synonyms that fit into the context, that is, without changing the meaning of the original text. Because retaining the meaning of the original text is a strict requirement, complex words need to be replaced with semantically similar words, whereas replacements with semantically related words (e.g., replacing *"pilot"* with *"airplane"* in *"Ferrari's pilot won the race"*) produce incorrect text which is more difficult to comprehend.

Simplification Using Distributional Vectors. We use the LIGHT-LS lexical simplification algorithm of Glavaš and Štajner (2015) which makes the word replacement decisions primarily based on semantic similarities between words in a distributional vector space.[8] For each word in the input text LIGHT-LS retrieves most similar replacement candidates from the vector space. The candidates are then ranked according to several measures of simplicity and fitness for the context. Finally, the replacement is made if the top-ranked candidate is estimated to be simpler than the original word. By plugging-in vector spaces specialized by the ER model into LIGHT-LS, we hope to generate true synonymous candidates more frequently than with the unspecialized distributional space.

Evaluation Setup. We evaluate LIGHT-LS on the LS dataset crowdsourced by Horn et al. (2014). For each indicated complex word Horn et al. (2014) collected 50 manual simplifications. We use two evaluation metrics from prior work (Horn et al., 2014; Glavaš and Štajner, 2015) to quantify the quality and frequency of word replacements: (1) *accurracy* (A) is the number of correct simplifications made (i.e., when the replacement made by the system is found in the list of manual replacements) divided by the total number of indicated complex words; and (2) *change* (C) is the percentage of indicated complex words that were replaced by the system (regardless of whether the replacement was correct). We plug into LIGHT-LS both unspecialized and specialized variants of three previously used English embedding spaces: GLOVE-CC, FASTTEXT, and SGNS-W2. Additionally, we again evaluate specializations of the same spaces produced by the state-of-the-art local retrofitting model ATTRACT-REPEL (Mrkšić et al., 2017).

[8]The Light-LS implementation is available at: https://bitbucket.org/gg42554/embesimp

	GLOVE-CC		FASTTEXT		SGNS-W2	
Emb. space	A	C	A	C	A	C
Distributional	66.0	**94.0**	57.8	84.0	56.0	79.1
Specialized						
ATTRACT-REPEL	67.6	87.0	69.8	89.4	64.4	86.7
ER-CNT	**73.8**	93.0	**71.2**	**93.2**	**68.4**	**92.3**

Table 4: Lexical simplification performance with explicit retrofitting applied on three input spaces.

Results and Analysis. The results with LIGHT-LS are summarized in Table 4. ER-CNT model yields considerable gains over unspecialized spaces for both metrics. This suggests that the ER-specialized embedding spaces allow LIGHT-LS to generate true synonymous candidate replacements more often than with unspecialized spaces, and also verifies the importance of specialization for the LS task. Our ER-CNT model now also yields better results than ATTRACT-REPEL in a real-world downstream task. Only 59.6 % of all indicated complex words and manual replacement candidates from the LS dataset are now covered by the linguistic constraints. This accentuates the need to specialize the full distributional space in downstream applications as done by the ER model, while ATTRACT-REPEL is limited to local vector updates only of words seen in the constraints. By learning a global specialization function the proposed ER models seem more resilient to the observed drop in coverage of test words by linguistic constraints. Table 5 shows example substitutions of LIGHT-LS when using different embedding spaces: original GLOVE-CC space and its specializations obtained with ER-CNT and ATTRACT-REPEL.

5.3.2 Dialog State Tracking

Finally, we also evaluate the importance of explicit retrofitting in a downstream language understand-

Text	GLOVE-CC	ATTRACT-REPEL	ER-CNT
*Wrestlers portrayed a **villain** or a hero as they followed a series of events that built tension*	*character*	*protagonist*	*demon*
*This large version number jump was due to a feeling that a version 1.0 with no major missing **pieces** was imminent.*	*ones*	*songs*	*parts*
*The storm continued, crossing North Carolina , and **retained** its strength until June 20 when it became extratropical near Newfoundland*	*lost*	*preserved*	*preserved*
*Tibooburra has an **arid**, desert climate with temperatures soaring above 40 Celsius in summer, often reaching as high as 47 degrees Celsius.*	*subtropical*	*humid*	*dry*

Table 5: Examples of lexical simplifications performed with the Light-LS tool when using different embedding spaces. The target word to be simplified is in bold.

GLOVE-CC embedding vectors	JGA
Distributional ($\mathbf{X}$)	.797
Specialized ($\mathbf{X}' = f(\mathbf{X})$)	
ATTRACT-REPEL	.817
ER-CNT	.816

Table 6: DST performance of GLOVE-CC embeddings specialized using explicit retrofitting.

ing task, namely dialog state tracking (DST) (Henderson et al., 2014; Williams et al., 2016). A DST model is typically the first component of a dialog system pipeline (Young, 2010), tasked with capturing user's goals and updating the dialog state at each dialog turn. Similarly as in lexical simplification, discerning similarity from relatedness is crucial in DST (e.g., a dialog system should not recommend an *"expensive pub in the south"* when asked for a *"cheap bar in the east"*).

Evaluation Setup. To evaluate the impact of specialized word vectors on DST, we employ the Neural Belief Tracker (NBT), a DST model that makes inferences purely based on pre-trained word vectors (Mrkšić et al., 2017).[9] NBT composes word embeddings into intermediate utterance and context representations. For full model details, we refer the reader to the original paper. Following prior work, our DST evaluation is based on the Wizard-of-Oz (WOZ) v2.0 dataset (Wen et al., 2017; Mrkšić et al., 2017) which contains 1,200 dialogs (600 training, 200 validation, and 400 test dialogs). We evaluate performance of the distributional and specialized GLOVE-CC embeddings and report it in terms of *joint goal accuracy* (JGA), a standard DST evaluation metric. All reported results are averages over 5 runs of the NBT model.

Results. We show DST performance in Table 6. The DST results tell a similar story like word similarity and lexical simplification results – the ER model substantially improves over the distributional space. With linguistic specialization constraints covering 57% of words from the WOZ dataset, ER model's performance is on a par with the ATTRACT-REPEL specialization. This further confirms our hypothesis that the importance of learning a global specialization for the full vocabulary in downstream tasks grows with the drop of the test word coverage by specialization constraints.

[9] https://github.com/nmrksic/neural-belief-tracker

6 Conclusion

We presented a novel method for specializing word embeddings to better discern similarity from other types of semantic relatedness. Unlike existing retrofitting models, which directly update vectors of words from external constraints, we use the constraints as training examples to learn an explicit specialization function, implemented as a deep feed-forward neural network. Our global specialization approach resolves the well-known inability of retrofitting models to specialize vectors of words unseen in the constraints. We demonstrated the effectiveness of the proposed model on word similarity benchmarks, and in two downstream tasks: lexical simplification and dialog state tracking. We also showed that it is possible to transfer the specialization to languages without linguistic constraints.

In future work, we will investigate explicit retrofitting methods for asymmetric relations like hypernymy and meronymy. We also intend to apply the method to other downstream tasks and to investigate the zero-shot language transfer of the specialization function for more language pairs.

ER code is publicly available at: `https://github.com/codogogo/explirefit`.

Acknowledgments

Ivan Vulić is supported by the ERC Consolidator Grant LEXICAL (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.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of ACL*, pages 86–90.

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.

Danushka Bollegala, Mohammed Alsuhaibani, Takanori Maehara, and Ken-ichi Kawarabayashi. 2016. Joint word representation learning using a corpus and a semantic lexicon. In *Proceedings of AAAI*, pages 2690–2696.

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.

Ryan Cotterell, Hinrich Schütze, and Jason Eisner. 2016. Morphological smoothing and extrapolation of word embeddings. In *Proceedings of ACL*, pages 1651–1660.

Paramveer S. Dhillon, Dean P. Foster, and Lyle H. Ungar. 2015. Eigenwords: Spectral word embeddings. *Journal of Machine Learning Research*, 16:3035–3078.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *Proceedings of ICLR: Workshop Papers*.

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.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of NAACL-HLT*, pages 634–643.

Christiane Fellbaum. 1998. *WordNet*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

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.

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

Matthew Henderson, Blaise Thomson, and Jason D. Wiliams. 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.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using wikipedia. In *Proceedings of the ACL*, pages 458–463.

Sujay Kumar Jauhar, Chris Dyer, and Eduard H. Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proceedings of NAACL*, pages 683–693.

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, Marie-Catherine de Marneffe, and Eric Fosler-Lussier. 2016a. Adjusting word embeddings with semantic intensity orders. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 62–69.

Joo-Kyung Kim, Gokhan Tur, Asli Celikyilmaz, Bin Cao, and Ye-Yi Wang. 2016b. 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.

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. 2014a. Dependency-based word embeddings. In *Proceedings of ACL*, pages 302–308.

Omer Levy and Yoav Goldberg. 2014b. 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.

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.

Nikola Ljubešić and Tomaž Erjavec. 2011. hrWaC and slWaC: Compiling web corpora for croatian and slovene. In *Proceedings of TSD*, pages 395–402.

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. 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.

Saif M. Mohammad, Bonnie J. Dorr, Graeme Hirst, and Peter D. Turney. 2013. Computing lexical contrast. *Computational Linguistics*, 39(3):555–590.

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, 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.

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.

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.

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.

Samuel L. Smith, David H.P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of ICLR (Conference Track)*.

Ivan Vulić and Nikola Mrkšić. 2017. Specialising word vectors for lexical entailment. *CoRR*, abs/1710.06371.

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*.

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.

Wen-tau Yih, Geoffrey Zweig, and John C. Platt. 2012. Polarity inducing latent semantic analysis. In *EMNLP-CoNLL*, pages 1212–1222.

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.