Specialising Word Vectors for Lexical Entailment

Ivan Vulić and **Nikola Mrkšić**
University of Cambridge
{iv250,nm480}@cam.ac.uk

Abstract

We present LEAR (**L**exical **E**ntailment **A**ttract-**R**epel), a novel post-processing method that transforms any input word vector space to emphasise the asymmetric relation of *lexical entailment* (LE), also known as the IS-A or hyponymy-hypernymy relation. By injecting external linguistic constraints (e.g., WordNet links) into the initial vector space, the LE specialisation procedure brings true hyponymy-hypernymy pairs closer together in the transformed Euclidean space. The proposed asymmetric distance measure adjusts the norms of word vectors to reflect the actual WordNet-style hierarchy of concepts. Simultaneously, a joint objective enforces semantic similarity using the symmetric cosine distance, yielding a vector space specialised for both lexical relations at once. LEAR specialisation achieves state-of-the-art performance in the tasks of hypernymy directionality, hypernymy detection and graded lexical entailment, demonstrating the effectiveness and robustness of the proposed model.

1 Introduction

Word representation learning has become a research area of central importance in NLP, with its usefulness demonstrated across application areas such as parsing (Chen and Manning, 2014), machine translation (Zou et al., 2013), and many others (Turian et al., 2010; Collobert et al., 2011). Standard techniques for inducing word embeddings rely on the *distributional hypothesis* (Harris, 1954), using co-occurrence information from large textual corpora to learn meaningful word representations (Mikolov et al., 2013; Levy and Goldberg, 2014; Pennington et al., 2014; Bojanowski et al., 2017).

A major drawback of the distributional hypothesis is that it coalesces different relationships, such as synonymy and topical relatedness, into a single vector space. A popular solution is to go beyond stand-alone unsupervised learning and fine-tune distributional vector spaces by using external knowledge from human- or automatically-constructed knowledge bases. This is often done as a *post-processing* step, where distributional vectors are gradually refined to satisfy linguistic constraints extracted from lexical resources such as WordNet (Faruqui et al., 2015; Mrkšić et al., 2016), the Paraphrase Database (PPDB) (Wieting et al., 2015), or BabelNet (Mrkšić et al., 2017; Vulić et al., 2017a). One advantage of post-processing methods is that they treat the input vector space as a *black box*, making them applicable to any input space.

A key property of these methods is their ability to transform the vector space by *specialising* it for a particular relationship between words.[1] Prior work has predominantly focused on distinguishing between semantic similarity and conceptual relatedness (Faruqui et al., 2015; Mrkšić et al., 2017; Vulić et al., 2017b). In this paper, we introduce a novel post-processing model which specialises vector spaces for the *lexical entailment* (LE) relation.

Word-level lexical entailment is an *asymmetric* semantic relation (Collins and Quillian, 1972; Beckwith et al., 1991). It is a key principle determining the organization of semantic networks into hierarchical structures such as semantic ontologies (Fellbaum, 1998). Automatic reasoning about LE supports tasks such as taxonomy creation (Snow et al., 2006; Navigli et al., 2011), natural language inference (Dagan et al., 2013; Bowman et al., 2015), text generation (Biran and McKeown, 2013), and metaphor detection (Mohler et al., 2013).

[1] Distinguishing between synonymy and antonymy has a positive impact on real-world language understanding tasks such as Dialogue State Tracking (Mrkšić et al., 2017).

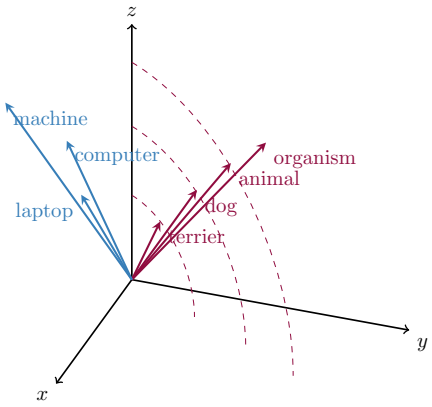


Figure 1: An illustration of LEAR specialisation. LEAR controls the arrangement of vectors in the transformed vector space by: **1)** emphasising symmetric similarity of LE pairs through cosine distance (by enforcing small angles between $\overrightarrow{terrier}$ and $\overrightarrow{dog}$ or $\overrightarrow{dog}$ and $\overrightarrow{animal}$); and **2)** by imposing an LE ordering using vector norms, adjusting them so that higher-level concepts have larger norms (e.g., $|\overrightarrow{animal}| > |\overrightarrow{dog}| > |\overrightarrow{terrier}|$).

Our novel LE specialisation model, termed LEAR (**L**exical **E**ntailment **A**ttract-**R**epel), is inspired by ATTRACT-REPEL, a state-of-the-art general specialisation framework (Mrkšić et al., 2017).[2] The key idea of LEAR, illustrated by Figure 1, is to pull desirable (ATTRACT) examples described by the constraints closer together, while at the same time pushing undesirable (REPEL) word pairs away from each other. Concurrently, LEAR (re-)arranges vector norms so that norm values in the Euclidean space reflect the hierarchical organization of concepts according to the given LE constraints: put simply, higher-level concepts are assigned larger norms. Therefore, LEAR simultaneously captures the hierarchy of concepts (through vector norms) and their similarity (through their cosine distance). The two pivotal pieces of information are combined into an *asymmetric distance measure* which quantifies the LE strength in the specialised space.

After specialising four well-known input vector spaces with LEAR, we test them in three standard word-level LE tasks (Kiela et al., 2015): **1)** hypernymy *directionality*; **2)** hypernymy *detection*; and **3)** *combined* hypernymy detection/directionality. Our specialised vectors yield notable improvements over the strongest baselines for each task, with each input space, demonstrating the effectiveness and robustness of LEAR specialisation.

[2] https://github.com/nmrksic/attract-repel

The employed asymmetric distance allows one to make graded assertions about hierarchical relationships between concepts in the specialised space. This property is evaluated using HyperLex, a recent *graded LE* dataset (Vulić et al., 2017). The LEAR-specialised vectors push state-of-the-art Spearman's correlation from 0.540 to 0.686 on the full dataset (2,616 word pairs), and from 0.512 to 0.705 on its noun subset (2,163 word pairs).

2 Methodology

2.1 The ATTRACT-REPEL Framework

Let V be the vocabulary, A the set of ATTRACT word pairs (e.g., *intelligent* and *brilliant*), and R the set of REPEL word pairs (e.g., *vacant* and *occupied*). The ATTRACT-REPEL procedure operates over mini-batches of such pairs $\mathcal{B}_A$ and $\mathcal{B}_R$. For ease of notation, let each word pair (x_l, x_r) in these two sets correspond to a vector pair $(\mathbf{x}_l, \mathbf{x}_r)$, so that a mini-batch of k_1 word pairs is given by $\mathcal{B}_A = [(\mathbf{x}_l^1, \mathbf{x}_r^1), \ldots, (\mathbf{x}_l^{k_1}, \mathbf{x}_r^{k_1})]$ (similarly for $\mathcal{B}_R$, which consists of k_2 example pairs).

Next, the sets of pseudo-negative examples $T_A = [(\mathbf{t}_l^1, \mathbf{t}_r^1), \ldots, (\mathbf{t}_l^{k_1}, \mathbf{t}_r^{k_1})]$ and $T_R = [(\mathbf{t}_l^1, \mathbf{t}_r^1), \ldots, (\mathbf{t}_l^{k_2}, \mathbf{t}_r^{k_2})]$ are defined as pairs of *negative examples* for each ATTRACT and REPEL example 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 ATTRACT 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 $\mathbf{x}_r$. Similarly, for each REPEL pair $(\mathbf{x}_l, \mathbf{x}_r)$, the negative example pair $(\mathbf{t}_l, \mathbf{t}_r)$ is chosen from the remaining in-batch vectors so that $\mathbf{t}_l$ is the vector furthest away from $\mathbf{x}_l$ and $\mathbf{t}_r$ is furthest from $\mathbf{x}_r$.

The negative examples are used to: **a)** force ATTRACT pairs to be closer to each other than to their respective negative examples; and **b)** to force REPEL pairs to be further away from each other than from their negative examples. The first term of the cost function pulls ATTRACT pairs together:

$$Att(\mathcal{B}_A, T_A) = \sum_{i=1}^{k_1} \Big[\tau\left(\delta_{att} + \mathbf{x}_l^i \mathbf{t}_l^i - \mathbf{x}_l^i \mathbf{x}_r^i\right) + \tau\left(\delta_{att} + \mathbf{x}_r^i \mathbf{t}_r^i - \mathbf{x}_l^i \mathbf{x}_r^i\right) \Big]$$

where $\tau(x) = \max(0, x)$ is the hinge loss function and δ_{att} is the attract margin which determines how much closer these vectors should be to each other than to their respective negative examples.

The second part of the cost function pushes REPEL word pairs away from each other:

$$Rep(\mathcal{B}_R, T_R) = \sum_{i=1}^{k_2} \big[\tau \left(\delta_{rep} + \mathbf{x}_l^i \mathbf{x}_r^i - \mathbf{x}_l^i \mathbf{t}_l^i\right) + \tau \left(\delta_{rep} + \mathbf{x}_l^i \mathbf{x}_r^i - \mathbf{x}_r^i \mathbf{t}_r^i\right) \big]$$

In addition to these two terms, an additional regularisation term is used to *preserve* the abundance of high-quality semantic content present in the distributional vector space, as long as this information does not contradict the injected linguistic constraints. If $V(\mathcal{B})$ is the set of all word vectors present in the given mini-batch, then:

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

where λ_{reg} is the L2 regularization constant and $\widehat{\mathbf{x}}_i$ denotes the original (distributional) word vector for word x_i. The full ATTRACT-REPEL cost function is given by the sum of all three terms.

2.2 LEAR: Encoding Lexical Entailment

In this section, the ATTRACT-REPEL framework is extended to model lexical entailment jointly with (symmetric) semantic similarity. To do this, the method uses an additional source of external lexical knowledge: let L be the set of *directed* lexical entailment constraints such as *(corgi, dog)*, *(dog, animal)*, or *(corgi, animal)*, with lower-level concepts on the left and higher-level ones on the right (the source of these constraints will be discussed in Section 3). The optimisation proceeds in the same way as before, considering a mini-batch of LE pairs $\mathcal{B}_L$ consisting of k_3 word pairs standing in the (directed) lexical entailment relation.

Unlike symmetric similarity, lexical entailment is an asymmetric relation which encodes a hierarchical ordering between concepts. Inferring the direction of the entailment relation between word vectors requires the use of an asymmetric distance function. We define three different ones, all of which use the word vector's norms to impose an ordering between high- and low-level concepts:

$$D_1(\mathbf{x}, \mathbf{y}) = |\mathbf{x}| - |\mathbf{y}| \quad (1)$$

$$D_2(\mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x}| - |\mathbf{y}|}{|\mathbf{x}| + |\mathbf{y}|} \quad (2)$$

$$D_3(\mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x}| - |\mathbf{y}|}{\max(|\mathbf{x}|, |\mathbf{y}|)} \quad (3)$$

The lexical entailment term (for the j-th asymmetric distance, $j \in 1, 2, 3$) is defined as:

$$LE_j(\mathcal{B}_L) = \sum_{i=1}^{k_3} D_j(\mathbf{x}_i, \mathbf{y}_i) \quad (4)$$

The first distance serves as the baseline: it uses the word vectors' norms to order the concepts, that is to decide which of the words is likely to be the higher-level concept. In this case, the magnitude of the difference between the two norms determines the 'intensity' of the LE relation. This is potentially problematic, as this distance does not impose a limit on the vectors' norms. The second and third metric take a more sophisticated approach, using the ratios of the differences between the two norms and either: **a)** the sum of the two norms; or **b)** the larger of the two norms. In doing that, these metrics ensure that the cost function only considers the norms' ratios. This means that the cost function no longer has the incentive to increase word vectors' norms past a certain point, as the magnitudes of norm ratios grow in size much faster than the linear relation defined by the first distance function.

To model the semantic and the LE relations jointly, the LEAR cost function jointly optimises the four terms of the expanded cost function:

$$C(\mathcal{B}_A, T_A, \mathcal{B}_R, T_R, \mathcal{B}_L, T_L) = Att(\mathcal{B}_S, T_S) + \ldots + Rep(\mathcal{B}_A, T_A) + Reg(\mathcal{B}_A, \mathcal{B}_R, \mathcal{B}_L) + \ldots + Att(\mathcal{B}_L, T_L) + LE_j(B_L)$$

LE Pairs as ATTRACT Constraints The combined cost function makes use of the batch of lexical constraints $\mathcal{B}_L$ twice: once in the defined asymmetric cost function LE_j, and once in the symmetric ATTRACT term $Att(\mathcal{B}_L, T_L)$. This means that words standing in the lexical entailment relation are forced to be similar both in terms of cosine distance (via the symmetric ATTRACT term) and in terms of the asymmetric LE distance from Eq. (4).

Decoding Lexical Entailment The defined cost function serves to encode semantic similarity and LE relations in the same vector space. Whereas the similarity can be inferred from the standard cosine distance, the LEAR optimisation embeds lexical entailment as a combination of the symmetric ATTRACT term and the newly defined asymmetric LE_j cost function. Consequently, the metric used to determine whether two words stand in the LE relation must combine the two cost terms as well.

We define the LE *decoding* metric as:

$$I_{LE}(\mathbf{x}, \mathbf{y}) = dcos(\mathbf{x}, \mathbf{y}) + D_j(\mathbf{x}, \mathbf{y}) \quad (5)$$

where $dcos(\mathbf{x}, \mathbf{y})$ denotes the cosine distance. This decoding function combines the symmetric and the asymmetric cost term, in line with the combination of the two used to perform LEAR specialisation. In the evaluation, we show that combining the two cost terms has a synergistic effect, with both terms contributing to stronger performance across all LE tasks used for evaluation.

3 Experimental Setup

Starting Distributional Vectors To test the robustness of LEAR specialisation, we experiment with a variety of well-known, publicly available English word vectors: **1)** Skip-Gram with Negative Sampling (SGNS) (Mikolov et al., 2013) trained on the Polyglot Wikipedia (Al-Rfou et al., 2013) by Levy and Goldberg (2014); **2)** GLOVE Common Crawl (Pennington et al., 2014); **3)** CONTEXT2VEC (Melamud et al., 2016), which replaces CBOW contexts with contexts based on bidirectional LSTMs (Hochreiter and Schmidhuber, 1997); and **4)** FASTTEXT (Bojanowski et al., 2017), a SGNS variant which builds word vectors as the sum of their constituent character n-gram vectors.[3]

Linguistic Constraints We use three groups of linguistic constraints in the LEAR specialisation model, covering three different relation types which are all beneficial to the specialisation process: directed **1)** *lexical entailment* (LE) *pairs*; **2)** *synonymy pairs*; and **3)** *antonymy pairs*. Synonyms are included as symmetric ATTRACT pairs (i.e., the $\mathcal{B}_A$ pairs) since they can be seen as defining a trivial symmetric IS-A relation (Rei and Briscoe, 2014; Vulić et al., 2017). For a similar reason, antonyms are clear REPEL constraints as they anti-correlate with the LE relation.[4] Synonymy and antonymy constraints are taken from prior work (Zhang et al., 2014; Ono et al., 2015): they are extracted from WordNet (Fellbaum, 1998) and Roget (Kipfer, 2009). In total, we work with 1,023,082 synonymy pairs (11.7 synonyms per word on average) and 380,873 antonymy pairs (6.5 per word).[5]

As in prior work (Nguyen et al., 2017; Nickel and Kiela, 2017), LE constraints are extracted from the WordNet hierarchy, relying on the transitivity of the LE relation. This means that we include both direct and indirect LE pairs in our set of constraints (e.g., (*pangasius, fish*), (*fish, animal*), and (*pangasius, animal*)). We retained only noun-noun and verb-verb pairs, while the rest were discarded: the final number of LE constraints is 1,545,630.[6]

Training Setup We adopt the original ATTRACT-REPEL model setup without any fine-tuning. 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 batch sizes set to $k_1 = k_2 = k_3 = 128$ for faster convergence.

4 Results and Discussion

We test and analyse LEAR-specialised vector spaces in two standard word-level LE tasks used in prior work: hypernymy directionality and detection (Section 4.1) and graded LE (Section 4.2).

4.1 LE Directionality and Detection

The first evaluation uses three classification-style tasks with increased levels of difficulty. The tasks are evaluated on three datasets used extensively in the LE literature (Roller et al., 2014; Santus et al., 2014; Weeds et al., 2014; Shwartz et al., 2017; Nguyen et al., 2017), compiled into an integrated evaluation set by Kiela et al. (2015).[7]

The first task, LE directionality, is conducted on 1,337 LE pairs originating from the BLESS evaluation set (Baroni and Lenci, 2011). Given a true LE pair, the task is to predict the correct hypernym. With LEAR-specialised vectors this is achieved by simply comparing the vector norms of each concept in a pair: the one with the larger norm is the hypernym (see Figure 1).

The second task, LE detection, involves a binary classification on the WBLESS dataset (Weeds et al., 2014) which comprises 1,668 word pairs standing

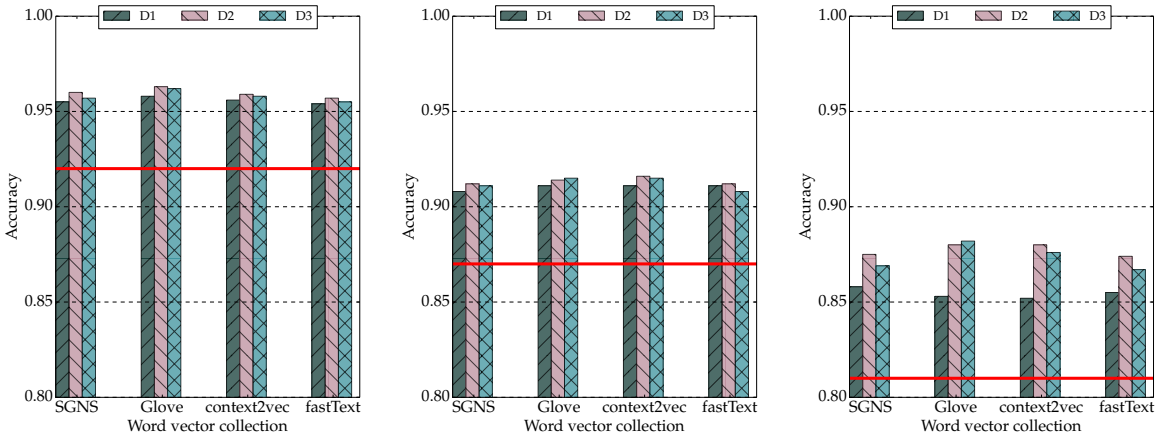
[3] All vectors are 300-dimensional except for the 600-dimensional CONTEXT2VEC vectors; for further details regarding the architectures and training setup of the used vector collections, we refer the reader to the original papers.

[4] In short, the question "*Is X a type of X?*" (synonymy) is trivially true, while the question "*Is ¬X a type of X?*" (antonymy) is trivially false.

[5] https://github.com/tticoin/AntonymDetection

[6] We also experimented with additional 30,491 LE constraints from the Paraphrase Database (PPDB) 2.0 (Pavlick et al., 2015). Adding them to the WordNet-based LE pairs makes no significant impact on the final performance. We also used synonymy and antonymy pairs from other sources, such as word pairs from PPDB used previously by Wieting et al. (2015), and BabelNet (Navigli and Ponzetto, 2012) used by Mrkšić et al. (2017), reaching the same conclusions.

[7] http://www.cl.cam.ac.uk/∼dk427/generality.html



(a) BLESS: Directionality **(b) WBLESS**: Detection **(c) BIBLESS**: Detect+direct

Figure 2: Summary of the results on three different word-level LE subtasks: (a) *directionality*; (b) *detection*; (c) *detection and directionality*. Vertical bars denote the results obtained by different input word vector spaces which are post-processed/specialised by our LEAR specialisation model using three variants of the asymmetric distance (D_1, D_2, D_3), see Section 2. Thick horizontal red lines refer to the best reported scores on each subtask for these datasets; the baseline scores are taken from Nguyen et al. (2017).

in a variety of relations (LE, meronymy-holonymy, co-hyponymy, reversed LE, no relation). The model has to detect a true LE pair, that is, to distinguish between the pairs where the statement *X is a (type of) Y* is true from all other pairs. With LEAR vectors, this classification is based on the asymmetric distance score: if the score is above a certain threshold, we classify the pair as "true LE", otherwise as "other". While Kiela et al. (2015) manually define the threshold value, we follow the approach of Nguyen et al. (2017) and cross-validate: in each of the 1,000 iterations, 2% of the pairs are sampled for threshold tuning, and the remaining 98% are used for testing. The reported numbers are therefore average accuracy scores.[8]

The final task, LE detection *and* directionality, concerns a three-way classification on BIBLESS, a relabeled version of WBLESS. The task is now to distinguish both LE pairs ($\rightarrow 1$) and reversed LE pairs ($\rightarrow -1$) from other relations ($\rightarrow 0$), and then additionally select the correct hypernym in each detected LE pair. We apply the same test protocol as in the LE detection task.

Results and Analysis The original paper of Kiela et al. (2015) reports the following best scores on each task: 0.88 (BLESS), 0.75 (WBLESS), 0.57 (BIBLESS). These scores were recently surpassed by Nguyen et al. (2017), who, instead of post-processing, combine WordNet-based constraints with an SGNS-style objective into a joint model. They report the best scores to date: 0.92 (BLESS), 0.87 (WBLESS), and 0.81 (BIBLESS).

The performance of the four LEAR-specialised word vector collections is shown in Figure 2 (together with the strongest baseline scores for each of the three tasks). The comparative analysis confirms the increased complexity of subsequent tasks. LEAR specialisation of *each* of the starting vector spaces consistently outperformed *all* baseline scores across *all* three tasks. The extent of the improvements is correlated with task difficulty: it is lowest for the easiest directionality task ($0.92 \rightarrow 0.96$), and highest for the most difficult detection plus directionality task ($0.81 \rightarrow 0.88$).

The results show that the two LEAR variants which do not rely on absolute norm values and perform a normalisation step in the asymmetric distance (D2 and D3) have a slight edge over the D1 variant which operates with unbounded norms. The difference in performance between D2/D3 and D1 is more pronounced in the graded LE task (see Section 4.2). This shows that the use of unbounded vector norms diminishes the importance of the symmetric cosine distance in the combined asymmetric distance. Conversely, the synergistic combination used in D2/D3 does not suffer from this issue.

The high scores achieved with each of the four

[8] We have conducted more LE directionality and detection experiments on other datasets such as EVALution (Santus et al., 2015), the $N_1 \vDash N_2$ dataset of Baroni et al. (2012), and the dataset of Lenci and Benotto (2012) with similar performances and findings. We do not report all these results for brevity and clarity of presentation.

	Norm		Norm		Norm
terrier	0.87	laptop	0.60	cabriolet	0.74
dog	2.64	computer	2.96	car	3.59
mammal	8.57	machine	6.15	vehicle	7.78
vertebrate	10.96	device	12.09	transport	8.01
animal	11.91	artifact	17.71	instrumentality	14.56
organism	20.08	object	23.55	–	–

Table 1: L2 norms for selected concepts from the WordNet hierarchy. Input: FASTTEXT; LEAR: D2.

word vector collections show that LEAR is not dependent on any particular word representation architecture. Moreover, the extent of the performance improvements in each task suggests that LEAR is able to reconstruct the concept hierarchy coded in the input linguistic constraints.

Further Discussion To verify that the knowledge concerning the position in the semantic hierarchy actually arises from vector norms, we also manually inspect the norms after LEAR specialisation. A few examples are provided in Table 1. They indicate a desirable pattern in the norm values which imposes a hierarchical ordering on the concepts. Note that the original distributional SGNS model (Mikolov et al., 2013) does not normalise vectors to unit length after training. However, these norms are not at all correlated with the desired hierarchical ordering, and are therefore useless for LE-related applications: the non-specialised distributional SGNS model scores 0.44, 0.48, and 0.34 on the three tasks, respectively.

4.2 Graded Lexical Entailment

Asymmetric distances in the LEAR-specialised space quantify the degree of lexical entailment between any two concepts. This means that they can be used to make fine-grained assertions regarding the hierarchical relationships between concepts. We test this property on HyperLex (Vulić et al., 2017), a gold standard dataset for evaluating how well word representation models capture graded LE, grounded in the notions of *concept (proto)typicality* (Rosch, 1973; Medin et al., 1984) and *category vagueness* (Kamp and Partee, 1995; Hampton, 2007) from cognitive science. HyperLex contains 2,616 word pairs (2,163 noun pairs and 453 verb pairs) scored by human raters in the $[0, 6]$ interval following the question *"To what degree is X a (type of) Y?"*[9]

As shown by the high inter-annotator agreement on HyperLex (0.85), humans are able to consistently reason about graded LE.[10] However, current state-of-the-art representation architectures are far from this ceiling. For instance, Vulić et al. (2017) evaluate a plethora of architectures and report a high-score of only 0.320 (see the summary table in Figure 3). Two recent representation models (Nickel and Kiela, 2017; Nguyen et al., 2017) focused on the LE relation in particular (and employing the same set of WordNet-based constraints as LEAR) report the highest score of 0.540 (on the entire dataset) and 0.512 (on the noun subset).

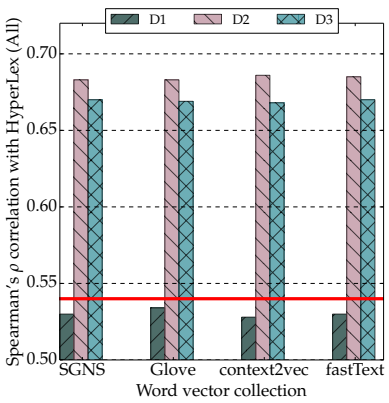
Results and Analysis We scored all HyperLex pairs using the combined asymmetric distance described by Equation (5), and then computed Spearman's rank correlation with the ground-truth ranking. Our results, together with the strongest baseline scores, are summarised in Figure 3.

The summary table in Figure 3(c) shows the HyperLex performance of several prominent LE models. We provide only a quick outline of these models here; further details can be found in the original papers. FREQ-RATIO exploits the fact that more general concepts tend to occur more frequently in textual corpora. SGNS (COS) uses non-specialised SGNS vectors and quantifies the LE strength using the symmetric cosine distance between vectors. A comparison of these models to the best-performing LEAR vectors shows the extent of the improvements achieved using the specialisation approach.

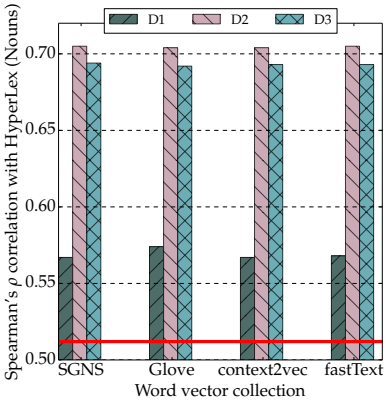
LEAR-specialised vectors also outperform SLQS-SIM (Santus et al., 2014) and VISUAL (Kiela et al., 2015), two LE detection models similar in spirit to LEAR. These models combine symmetric semantic similarity (through cosine distance) with an asymmetric measure of lexical generality obtained either from text (SLQS-SIM) or visual data (VISUAL). The results on HyperLex indicate that the two generality-based measures are too coarse-grained for graded LE judgements. These models were originally constructed to tackle LE directionality and detection tasks (see Section 4.1), but their performance is surpassed by LEAR on those tasks as well. The VISUAL model outperforms SLQS-SIM. However, its numbers on BLESS (0.88), WBLESS

[9] From another perspective, one might say that graded LE provides finer-grained human judgements on a continuous scale rather than simplifying the judgements into binary discrete decisions. For instance, the HyperLex score for the pair *(girl, person)* is 5.91/6, the score for *(guest, person)* is 4.33, while the score for the reversed pair *(person, guest)* is 1.73.

[10] For further details concerning HyperLex, we refer the reader to the resource paper (Vulić et al., 2017). The dataset is available at: *http://people.ds.cam.ac.uk/iv250/hyperlex.html*



(a) HyperLex: **All**



(b) HyperLex: **Nouns**

	All
FREQ-RATIO	0.279
SGNS (COS)	0.205
SLQS-SIM	0.228
VISUAL	0.209
WN-BEST	0.234
WORD2GAUSS	0.206
SIM-SPEC	0.320
ORDER-EMB	0.191
POINCARÉ (nouns)	0.512
HYPERVEC	0.540
Best LEAR	**0.686**

(c) Summary

Figure 3: Results on the graded LE task defined by HyperLex. Following Nickel and Kiela (2017), we use Spearman's rank correlation scores on: **a)** the entire dataset (2,616 noun and verb pairs); and **b)** its noun subset (2,163 pairs). The summary table shows the performance of other well-known architectures on the full HyperLex dataset, compared to the best results achieved using LEAR specialisation.

(0.75), and BIBLESS (0.57) are far from the top-performing LEAR vectors (0.96, 0.92, 0.88).

WN-BEST denotes the best result with asymmetric similarity measures which use the WordNet structure as their starting point (Wu and Palmer, 1994; Pedersen et al., 2004). The reported results suggest it is more effective to use WordNet as the source of constraints for specialisation models.

WORD2GAUSS (Vilnis and McCallum, 2015) represents words as multivariate K-dimensional Gaussians rather than points in the embedding space: it is therefore naturally asymmetric and was used in LE tasks before, but its performance on HyperLex indicates that it cannot effectively capture the subtleties required to model graded LE.

Most importantly, LEAR outperforms three recent (and conceptually different) architectures: ORDER-EMB (Vendrov et al., 2016), POINCARÉ (Nickel and Kiela, 2017), and HYPERVEC (Nguyen et al., 2017). Like LEAR, all of these models complement distributional knowledge with external linguistic constraints extracted from WordNet. Each model uses a different strategy to exploit the hierarchical relationships encoded in these constraints (their approaches are discussed in Section 5). However, LEAR, as the first LE-oriented post-processor, is able to utilise the constraints more effectively than its competitors. Another advantage of LEAR is its applicability to any input vector space.

Figures 3(a) and 3(b) indicate that the two LEAR variants which rely on norm ratios (D2 and D3), rather than on absolute (unbounded) norm differences (D1), achieve stronger performance on HyperLex. The highest correlation scores are again achieved by D2 with all input vector spaces.

	WBLESS	BIBLESS	HL-A	HL-N
LEAR variant				
SYM-ONLY	0.687	0.679	0.469	0.429
ASYM-ONLY	0.867	0.824	0.529	0.565
FULL	**0.912**	**0.875**	**0.686**	**0.705**

Table 2: Analysing the importance of the synergy in the FULL LEAR model on the final performance on WBLESS, BLESS, HyperLex-All (HL-A) and HyperLex-Nouns (HL-N). Input: FASTTEXT. D2.

Why Symmetric + Asymmetric? In another experiment, we analyse the contributions of both LE-related terms in the LEAR combined objective function (see Section 2.2). We compare three variants of LEAR: **1)** a symmetric variant which does not arrange vector norms using the $LE_j(\mathcal{B}_L)$ term (SYM-ONLY); **2)** a variant which arranges norms, but does not use LE constraints as additional symmetric ATTRACT constraints (ASYM-ONLY); and **3)** the full LEAR model, which uses both cost terms (FULL). The results with one input space (similar results are achieved with others) are shown in Table 2. This table shows that, while the stand-alone ASYM-ONLY term seems more beneficial than the SYM-ONLY one, using the two terms jointly yields the strongest performance across all LE tasks.

LE and Semantic Similarity We also test whether the asymmetric LE term harms the (norm-independent) cosine distances used to represent semantic similarity. The LEAR model is compared

to the original ATTRACT-REPEL model making use of the same set of linguistic constraints. Two true semantic similarity datasets are used for evaluation: SimLex-999 (Hill et al., 2015) and SimVerb-3500 (Gerz et al., 2016). There is no significant difference in performance between the two models, both of which yield similar results on SimLex (Spearman's rank correlation of ≈ 0.71) and SimVerb (≈ 0.70). This proves that cosine distances remain preserved during the optimization of the asymmetric objective performed by the joint LEAR model.

5 Related Work

Word Vectors and Lexical Entailment Since the hierarchical LE relation is one of the fundamental building blocks of semantic taxonomies and hierarchical concept categorisations (Beckwith et al., 1991; Fellbaum, 1998), a significant amount of research in semantics has been invested into its automatic detection and classification. Early work relied on asymmetric directional measures (Weeds et al., 2004; Clarke, 2009; Kotlerman et al., 2010; Lenci and Benotto, 2012, i.a.) which were based on the distributional inclusion hypothesis (Geffet and Dagan, 2005) or the distributional informativeness or generality hypothesis (Herbelot and Ganesalingam, 2013; Santus et al., 2014). However, these approaches have recently been superseded by methods based on word embeddings. These methods build dense real-valued vectors for capturing the LE relation, either directly in the LE-focused space (Vilnis and McCallum, 2015; Vendrov et al., 2016; Henderson and Popa, 2016; Nickel and Kiela, 2017; Nguyen et al., 2017) or by using the vectors as features for supervised LE detection models (Tuan et al., 2016; Shwartz et al., 2016; Nguyen et al., 2017; Glavaš and Ponzetto, 2017).

Several LE models embed useful hierarchical relations from external resources such as WordNet into LE-focused vector spaces, with solutions coming in different flavours. The model of Yu et al. (2015) is a dynamic distance-margin model optimised for the LE detection task using hierarchical WordNet constraints. This model was extended by Tuan et al. (2016) to make use of contextual sentential information. A major drawback of both models is their inability to make directionality judgements. Further, their performance has recently been surpassed by the HYPERVEC model of Nguyen et al. (2017). This model combines WordNet constraints with the SGNS distributional objective into a joint model. As such, the model is tied to the SGNS objective and any change of the distributional modelling paradigm implies a change of the entire HYPERVEC model. This makes their model less versatile than the proposed LEAR framework. Moreover, the results achieved using LEAR specialisation achieve substantially better performance across all LE tasks used for evaluation.

Another model similar in spirit to LEAR is the ORDER-EMB model of Vendrov et al. (2016), which encodes hierarchical structure by imposing a partial order in the embedding space: higher-level concepts get assigned higher per-coordinate values in a d-dimensional vector space. The model minimises the violation of the per-coordinate orderings during training by relying on hierarchical WordNet constraints between word pairs. Finally, the POINCARÉ model of Nickel and Kiela (2017) makes use of hyperbolic spaces to learn general-purpose LE embeddings based on n-dimensional Poincaré balls which encode both hierarchy and semantic similarity, again using the WordNet constraints. In this paper, we demonstrate that LE-specialised word embeddings with stronger performance can be induced using a simpler model operating in more intuitively interpretable Euclidean vector spaces.

6 Conclusion and Future Work

This paper proposed LEAR, a vector space specialisation procedure which simultaneously injects symmetric and asymmetric constraints into existing vector spaces, performing joint specialisation for two properties: *lexical entailment* and *semantic similarity*. Since the former is not symmetric, LEAR uses an asymmetric cost function which encodes the hierarchy between concepts by manipulating the norms of word vectors, assigning higher norms to higher-level concepts. Specialising the vector space for both relations has a synergistic effect: LEAR-specialised vectors attain state-of-the-art performance in judging semantic similarity and set new high scores across four different lexical entailment tasks. The code for the LEAR model is available from: `github.com/nmrksic/lear`.

In future work, we plan to apply a similar methodology to other asymmetric relations (e.g., *meronymy*), as well as to investigate fine-grained models which can account for differing path lengths from the WordNet hierarchy. Porting the model to other languages and enabling cross-lingual applications is another future direction.

Acknowledgments

IV 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.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of EACL*, pages 23–32.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop*, pages 1–10.

Richard Beckwith, Christiane Fellbaum, Derek Gross, and George A. Miller. 1991. WordNet: A lexical database organized on psycholinguistic principles. *Lexical acquisition: Exploiting on-line resources to build a lexicon*, pages 211–231.

Or Biran and Kathleen McKeown. 2013. Classifying taxonomic relations between pairs of Wikipedia articles. In *Proceedings of IJCNLP*, pages 788–794.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the ACL*, 5:135–146.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, pages 632–642.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*, pages 740–750.

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

Allan M. Collins and Ross M. Quillian. 1972. Experiments on semantic memory and language comprehension. *Cognition in Learning and Memory*.

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.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

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*.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of ACL*, pages 107–114.

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.

James A. Hampton. 2007. Typicality, graded membership, and vagueness. *Cognitive Science*, 31(3):355–384.

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

James Henderson and Diana Popa. 2016. A vector space for distributional semantics for entailment. In *Proceedings of ACL*, pages 2052–2062.

Aurélie Herbelot and Mohan Ganesalingam. 2013. Measuring semantic content in distributional vectors. In *Proceedings of ACL*, pages 440–445.

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.

Hans Kamp and Barbara Partee. 1995. Prototype theory and compositionality. *Cognition*, 57(2):129–191.

Douwe Kiela, Laura Rimell, Ivan Vulić, and Stephen Clark. 2015. Exploiting image generality for lexical entailment detection. In *Proceedings of ACL*, pages 119–124.

Barbara Ann Kipfer. 2009. *Roget's 21st Century Thesaurus (3rd Edition)*. Philip Lief Group.

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

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In *Proceedings of *SEM*, pages 75–79.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of ACL*, pages 302–308.

Douglas L. Medin, Mark W. Altom, and Timothy D. Murphy. 1984. Given versus induced category representations: Use of prototype and exemplar information in classification. *Journal of Experimental Psychology*, 10(3):333–352.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. Context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of CoNLL*, pages 51–61.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Michael Mohler, David Bracewell, Marc Tomlinson, and David Hinote. 2013. Semantic signatures for example-based linguistic metaphor detection. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 27–35.

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.

Roberto Navigli, Paola Velardi, and Stefano Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proceedings of IJCAI*, pages 1872–1877.

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.

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.

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.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity - Measuring the relatedness of oncepts. In *Proceedings of AAAI*, pages 1024–1025.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Marek Rei and Ted Briscoe. 2014. Looking for hyponyms in vector space. In *Proceedings of CoNLL*, pages 68–77.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING*, pages 1025–1036.

Eleanor H. Rosch. 1973. Natural categories. *Cognitive Psychology*, 4(3):328–350.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of EACL*, pages 38–42.

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. EVALution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 64–69.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of ACL*, pages 2389–2398.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In *Proceedings of EACL*, pages 65–75.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of ACL*, pages 801–808.

Luu Anh Tuan, Yi Tay, Siu Cheung Hui, and See Kiong Ng. 2016. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of EMNLP*, pages 403–413.

Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-embeddings of images and language. In *Proceedings of ICLR (Conference Track)*.

Luke Vilnis and Andrew McCallum. 2015. Word representations via Gaussian embedding. In *Proceedings of ICLR (Conference Track)*.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*.

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.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING*, pages 2249–2259.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of COLING*, pages 1015–1021.

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.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of ACL*, pages 133–138.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning term embeddings for hypernymy identification. In *Proceedings of IJCAI*, pages 1390–1397.

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.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of EMNLP*, pages 1393–1398.