Generalized Tuning of Distributional Word Vectors for Monolingual and Cross-Lingual Lexical Entailment

Goran Glavaš
University of Mannheim
Data and Web Science Group
B6, 26, DE-68161 Mannheim, Germany
goran@informatik.uni-mannheim.de

Ivan Vulić
PolyAI Ltd.
144A Clerkenwell Road
London, United Kingdom
ivan@poly-ai.com

Abstract

Lexical entailment (LE; also known as *hyponymy-hypernymy* or *is-a* relation) is a core asymmetric lexical relation that supports tasks like taxonomy induction and text generation. In this work, we propose a simple and effective method for fine-tuning distributional word vectors for LE. Our *Generalized Lexical ENtailment* model (GLEN) is decoupled from the word embedding model and applicable to any distributional vector space. Yet – unlike existing retrofitting models – it captures a general specialization function allowing for LE-tuning of the entire distributional space and not only the vectors of words seen in lexical constraints. Coupled with a multilingual embedding space, GLEN seamlessly enables cross-lingual LE detection. We demonstrate the effectiveness of GLEN in graded LE and report large improvements (over 20% in accuracy) over state-of-the-art in cross-lingual LE detection.

1 Background and Motivation

Lexical entailment (LE; *hyponymy-hypernymy* or *is-a* relation), is a fundamental asymmetric lexico-semantic relation (Collins and Quillian, 1972; Beckwith et al., 1991) and a key building block of lexico-semantic networks and knowledge bases (Fellbaum, 1998; Navigli and Ponzetto, 2012). Reasoning about word-level entailment supports a multitude of tasks such as taxonomy induction (Snow et al., 2006; Navigli et al., 2011; Gupta et al., 2017), natural language inference (Dagan et al., 2013; Bowman et al., 2015; Williams et al., 2018), metaphor detection (Mohler et al., 2013), and text generation (Biran and McKeown, 2013).

Due to their distributional nature (Harris, 1954), embedding models (Mikolov et al., 2013; Levy and Goldberg, 2014; Pennington et al., 2014; Melamud et al., 2016; Bojanowski et al., 2017; Peters et al., 2018, *inter alia*) conflate paradigmatic relations (e.g., synonymy, antonymy, LE, meronymy) and the broader topical (i.e., syntagmatic) relatedness (Schwartz et al., 2015; Mrkšić et al., 2017). Consequently, distributional vectors (i.e., embeddings) cannot be directly used to reliably detect LE.

Embedding *specialization* methods remedy for the semantic vagueness of distributional spaces, forcing the vectors to conform to external linguistic constraints (e.g., synonymy or LE word pairs) in order to emphasize the lexico-semantic relation of interest (e.g., semantic similarity of LE) and diminish the contributions of other types of semantic association. Lexical specialization models generally belong to one of the two families: (1) *joint optimization models* and (2) *retrofitting* (also known as *fine-tuning* or *post-processing*) models. Joint models incorporate linguistic constraints directly into the objective of an embedding model, e.g., Skip-Gram (Mikolov et al., 2013), by modifying the prior or regularization of the objective (Yu and Dredze, 2014; Xu et al., 2014; Kiela et al., 2015) or by augmenting the objective with additional factors reflecting linguistic constraints (Ono et al., 2015; Osborne et al., 2016; Nguyen et al., 2017). Joint models are tightly coupled to a concrete embedding model – any modification to the underlying embedding models warrants a modification of the whole joint model, along with the expensive retraining. Conversely, retrofitting models (Faruqui et al., 2015; Wieting et al., 2015; Nguyen et al., 2016; Mrkšić et al., 2017; Vulić and Mrkšić, 2018, *inter alia*) change the distributional spaces post-hoc, by fine-tuning word vectors so that they conform to external linguistic constraints. Advantageously, this makes retrofitting models more flexible, as they can be applied to any pre-trained distributional space. On the downside, retrofitting models specialize only the vectors of words *seen* in constraints, leaving vectors of *unseen* words unchanged.

In this work, we propose an LE-specialization framework that combines the strengths of both

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4824–4830
Florence, Italy, July 28 - August 2, 2019.

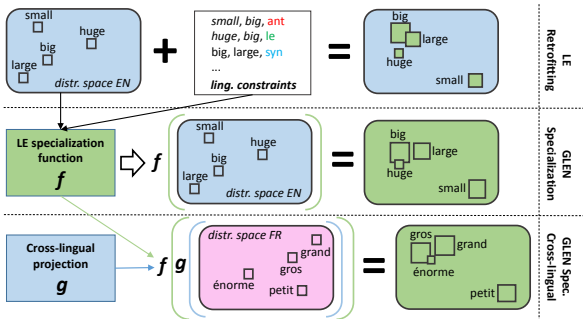

Figure 1: High-level illustration of GLEN. Row #1: LE-retrofitting – specializes only vectors of constraint words (from language $L1$); Row #2: GLEN – learns the specialization function f using constraints (from $L1$) as supervision; Row #3: Cross-lingual GLEN: LE-tuning of vectors from language $L2$ – f applied to $L2$ vectors projected (function g) to the $L1$ embedding space.

model families: unlike joint models, our generalized LE specialization (dubbed GLEN) is easily applicable to any embedding space. Yet, unlike the retrofitting models, it LE-specializes the entire distributional space and not just the vectors of words from external constraints. GLEN utilizes linguistic constraints as training examples in order to learn a general LE-specialization function (instantiated simply as a feed-forward neural net), which can then be applied to the entire distributional space. The difference between LE-retrofitting and GLEN is illustrated in Figure 1. Moreover, with GLEN's ability to LE-specialize unseen words we can seamlessly LE-specialize word vectors of another language ($L2$), assuming we previously project them to the distributional space of $L1$ for which we had learned the specialization function. To this end, we can leverage any from the plethora of resource-lean methods for learning the cross-lingual projection (function g in Figure 1) between monolingual distributional vector spaces (Smith et al., 2017; Conneau et al., 2018; Artetxe et al., 2018, *inter alia*).[1]

Conceptually, GLEN is similar to the explicit retrofitting model of Glavaš and Vulić (2018), who focus on the symmetric semantic similarity relation. In contrast, GLEN has to account for the asymmetric nature of the LE relation. Besides joint (Nguyen et al., 2017) and retrofitting (Vulić and Mrkšić, 2018) models for LE, there is a number of supervised LE detection models that employ distributional vectors as input features (Tuan et al., 2016; Shwartz et al., 2016; Glavaš and Ponzetto, 2017; Rei et al., 2018). These models, however, predict LE for pairs of words, but do not produce LE-specialized word vectors, which are directly pluggable into downstream models.

[1] See (Ruder et al., 2018b; Glavaš et al., 2019) for a comprehensive overview of models for inducing cross-lingual word embedding spaces.

2 Generalized Lexical Entailment

Following LEAR (Vulić and Mrkšić, 2018), the state-of-the-art LE-retrofitting model, we use three types of linguistic constraints to learn the general specialization f: synonyms, antonyms, and LE (i.e., hyponym-hypernym) pairs. Similarity-focused specialization models tune only the direction of distributional vectors (Mrkšić et al., 2017; Glavaš and Vulić, 2018; Ponti et al., 2018). In LE-specialization we need to emphasize similarities but also reflect the hierarchy of concepts offered by LE relations (e.g., *car* should be similar to both *Ferrari* and *vehicle* but is a hyponym only of *vehicle*). GLEN learns a specialization function f that rescales vector norms in order to reflect the hierarchical LE relation. To this end, we use the following asymmetric distance between vectors defined in terms of their Euclidean norms:

$$d_N(\mathbf{x}_1, \mathbf{x}_2) = \frac{\|\mathbf{x}_1\| - \|\mathbf{x}_2\|}{\|\mathbf{x}_1\| + \|\mathbf{x}_2\|} \quad (1)$$

Simultaneously, GLEN aims to bring closer together in direction vectors for synonyms and LE pairs and to push vectors of antonyms further apart. We use the cosine distance d_C as a symmetric measure of direction (dis)similarity between vectors. We combine the asymmetric distance d_N and symmetric d_C in different objective functions that we optimize to learn the LE-specialization function f.

Lexical Constraints as Training Instances. For each constraint type – synonyms, antonyms, and LE pairs – we create separate batches of training instances. Let $\{\mathbf{x}_1^E, \mathbf{x}_2^E\}_K$, $\{\mathbf{x}_1^S, \mathbf{x}_2^S\}_K$, and $\{\mathbf{x}_1^A, \mathbf{x}_2^A\}_K$ be the batches of K LE, synonymy, and antonymy pairs, respectively. For each constraint $(\mathbf{x}_1, \mathbf{x}_2)$ we create a pair of *negative* vectors $(\mathbf{y}_1, \mathbf{y}_2)$ such that $\mathbf{y}_1$ is the vector within the batch (except $\mathbf{x}_2$), closest to $\mathbf{x}_1$ and $\mathbf{y}_2$ the vector closest to $\mathbf{x}_2$ (but not $\mathbf{x}_1$) in terms of some distance or similarity metric. For LE constraints, we find $\mathbf{y}_1$ and $\mathbf{y}_2$ that minimize $d_N(\mathbf{x}_1, \mathbf{y}_1) + d_C(\mathbf{x}_1, \mathbf{y}_1)$ and $d_N(\mathbf{y}_2, \mathbf{x}_2) + d_C(\mathbf{x}_2, \mathbf{y}_2)$, respectively. Intuitively, we want our model to predict a smaller LE distance $d_N + d_C$ for a positive LE pair $(\mathbf{x}_1, \mathbf{x}_2)$ than for negative pairs $(\mathbf{x}_1, \mathbf{y}_1)$ and $(\mathbf{x}_2, \mathbf{y}_2)$ in the specialized space. By choosing the most-challenging negative pairs, i.e., $\mathbf{y}_1$ and $\mathbf{y}_2$ that are respectively closest

to $\mathbf{x}_1$ and $\mathbf{x}_2$ in terms of LE distance in the distributional space, we force our model to learn a more robust LE specialization function (this is further elaborated in the description of the objective function). Analogously, for positive synonym pairs, $\mathbf{y}_1$ and $\mathbf{y}_2$ are the vectors closest to $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, but in terms of only the (symmetric) cosine distance d_C. Finally, for antonyms, $\mathbf{y}_1$ is the vector maximizing $d_C(\mathbf{x}_1, \mathbf{y}_1)$ and $\mathbf{y}_2$ the vector that maximizes $d_C(\mathbf{x}_2, \mathbf{y}_2)$. In this case, we want the vectors of antonyms $\mathbf{x}_1$ and $\mathbf{x}_2$ after specialization to be further apart from one another (according to d_C) than from, respectively, the vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ that are most distant to them in the original distributional space. A training batch, with K entailment (E), synonymy (S), or antonymy (A) instances, is obtained by coupling constraints $(\mathbf{x}_1, \mathbf{x}_2)$ with their negative vectors $(\mathbf{y}_1, \mathbf{y}_2)$: $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2\}_K$.

Specialization Function. The parametrized specialization function $f(\mathbf{x}; \theta) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ (with d being the embedding size), transforms the distributional space to the space that better captures the LE relation. Once we learn the specialization function f (i.e., we tune the parameters θ), we can LE-specialize the *entire* distributional embedding space $\mathbf{X}$ (i.e., the vectors of *all* vocabulary words): $\mathbf{X}' = f(\mathbf{X}; \theta)$. For simplicity, we define f to be a (fully-connected) feed-forward net with H hidden layers of size d_h and non-linear activation ψ. The i-th hidden layer ($i \in \{1, \ldots, H\}$) is parametrized by the weight matrix $\mathbf{W}^i$ and the bias vector $\mathbf{b}^i$:[2]

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

Objectives and Training. We define four losses which we combine into training objectives for different constraint types (E, S, and A). The asymmetric loss l_a forces the asymmetric margin-based distance d_N to be larger for negative pairs $(\mathbf{x}_1, \mathbf{y}_1)$ and $(\mathbf{y}_2, \mathbf{x}_2)$ than for the positive (true LE) pair $(\mathbf{x}_1, \mathbf{x}_2)$ by at least the margin δ_a:

$$\begin{aligned} l_a = &\sum_{k=1}^{K} \tau\left(\delta_a - d_N\Big(f(\mathbf{x}_1^k), f(\mathbf{y}_1^k)\Big) + d_N\Big(f(\mathbf{x}_1^k), f(\mathbf{x}_2^k)\Big)\right) \\ &+ \tau\left(\delta_a - d_N\Big(f(\mathbf{y}_2^k), f(\mathbf{x}_2^k)\Big) + d_N\Big(f(\mathbf{x}_1^k), f(\mathbf{x}_2^k)\Big)\right) \quad (3) \end{aligned}$$

where $\tau(x) = \max(0, x)$ is the ramp function. The similarity loss l_s pushes the vectors $\mathbf{x}_1$, and $\mathbf{x}_2$ to be direction-wise closer to each other than to negative vectors $\mathbf{y}_1$ and $\mathbf{y}_2$, by margin δ_s:

$$\begin{aligned} l_s = &\sum_{k=1}^{K} \tau\left(\delta_s - d_C\Big(f(\mathbf{x}_1^k), f(\mathbf{y}_1^k)\Big) + d_C\Big(f(\mathbf{x}_1^k), f(\mathbf{x}_2^k)\Big)\right) \\ &+ \tau\left(\delta_s - d_C\Big(f(\mathbf{x}_2^k), f(\mathbf{y}_2^k)\Big) + d_N\Big(f(\mathbf{x}_1^k), f(\mathbf{x}_2^k)\Big)\right) \quad (4) \end{aligned}$$

The dissimilarity loss l_d pushes vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ further away from each other than from respective negative vectors $\mathbf{y}_1$ and $\mathbf{y}_2$, by the margin δ_d:

$$\begin{aligned} l_d = &\sum_{k=1}^{K} \tau\left(\delta_d - d_C\Big(f(\mathbf{x}_1^k), f(\mathbf{x}_2^k)\Big) + d_C\Big(f(\mathbf{x}_1^k), f(\mathbf{y}_1^k)\Big)\right) \\ &+ \tau\left(\delta_d - d_C\Big(f(\mathbf{x}_1^k), f(\mathbf{x}_2^k)\Big) + d_N\Big(f(\mathbf{x}_2^k), f(\mathbf{y}_2^k)\Big)\right) \quad (5) \end{aligned}$$

We also define the regularization loss l_r, preventing f from destroying the useful semantic information contained in distributional vectors:

$$\begin{aligned} l_r = &\sum_{k=1}^{K} d_C\left(\mathbf{x}_1^k, f(\mathbf{x}_1^k)\right) + d_C\left(\mathbf{x}_2^k, f(\mathbf{x}_2^k)\right) \\ &+ d_C\left(\mathbf{y}_1^k, f(\mathbf{y}_1^k)\right) + d_C\left(\mathbf{y}_2^k, f(\mathbf{y}_2^k)\right). \quad (6) \end{aligned}$$

Finally, we define different objectives for different constraints types (E, S, and A):

$$\begin{aligned} J_E &= l_s(E) + \lambda_a \cdot l_a(E) + \lambda_r \cdot l_r(E); \\ J_S &= l_s(S) + \lambda_r \cdot l_r(S); \\ J_A &= l_d(A) + \lambda_r \cdot l_r(A), \quad (7) \end{aligned}$$

where λ_a and λ_r scale the contributions of the asymmetric and regularization losses, respectively. J_E pushes LE vectors to be similar in direction (loss l_s) and different in norm (loss l_a) after specialization. J_S forces vectors of synonyms to be closer together (loss l_s) and J_A vectors of antonyms to be further apart (loss l_d) in direction after specialization, both without affecting vector norms. We tune hyperparameters (δ_a, δ_s, δ_d, λ_a, and λ_r) via cross-validation, with train and validation portions containing randomly shuffled E, S, and A batches.

Inference. We infer the strength of the LE relation between vectors $\mathbf{x}'_1 = f(\mathbf{x}_1)$ and $\mathbf{x}'_2 = f(\mathbf{x}_2)$ with an asymmetric LE distance combining d_C and d_N: $I_{LE}(\mathbf{x}'_1, \mathbf{x}'_2) = d_C(\mathbf{x}'_1, \mathbf{x}'_2) + d_N(\mathbf{x}'_1, \mathbf{x}'_2)$. True LE pairs should have a small d_C and negative d_N. We thus rank LE candidate word pairs according to their I_{LE} scores, from smallest to largest. For the binary LE detection, I_{LE} is binarized via threshold t: if $I_{LE} < t$, we predict that LE holds.

Cross-Lingual (CL) LE Specialization. After learning the generalized LE-specialization function f, we can apply it to specialize any vector that comes from the same distributional vector space

[2] The 0-th "hidden layer" is the input distributional vector: $h^0(\mathbf{x}; \theta_0) = \mathbf{x}$ and $\theta_0 = \varnothing$, following the notation of Eq. (2).

that we used in training. Let L_1 be the language for which we have the linguistic constraints and let $\mathbf{X}_{L1}$ be its corresponding distributional space. Let $\mathbf{X}_{L2}$ be the distributional space of another language L_2. Assuming a function $g : \mathbb{R}^{d_{L2}} \rightarrow \mathbb{R}^{d_{L1}}$ that projects vectors from $\mathbf{X}_{L2}$ to $\mathbf{X}_{L1}$, we can straightforwardly LE-specialize the distributional space of L_2 by composing functions f and g: $\mathbf{X'}_{L2} = f\left(g(\mathbf{X}_{L2})\right)$. Recently, a large number of projection-based models have been proposed for inducing bilingual word embedding spaces (Smith et al., 2017; Conneau et al., 2018; Artetxe et al., 2018; Ruder et al., 2018a; Joulin et al., 2018, *inter alia*), most of them requiring limited (word-level) or no bilingual supervision. Based on a few thousand (manually created or automatically induced) word-translation pairs, these models learn a linear mapping $\mathbf{W}_g$ that projects the vectors from $\mathbf{X}_{L2}$ to the space $\mathbf{X}_{L1}$: $g(\mathbf{X}_{L2}) = \mathbf{X}_{L2}\mathbf{W}_g$. The cross-lingual space is then given as: $\mathbf{X}_{L1} \cup \mathbf{X}_{L2}\mathbf{W}_g$. Due to simplicity and robust downstream performance,[3] we opt for the simple supervised learning of the cross-lingual projection matrix $\mathbf{W}_g$ (Smith et al., 2017) based on (closed-form) solution of the Procrustes problem (Schönemann, 1966). Let $\mathbf{X}_S \subset \mathbf{X}_{L2}$ and $\mathbf{X}_T \subset \mathbf{X}_{L1}$ be the subsets of the two monolingual embedding spaces, containing (row-aligned) vectors of word translations. We then obtain the projection matrix as $\mathbf{W}_g = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is the singular value decomposition of the product matrix $\mathbf{X}_T{\mathbf{X}_S}^\top$.

3 Evaluation

Experimental Setup. We work with Wikipedia-trained FASTTEXT embeddings (Bojanowski et al., 2017). We take English constraints from previous work – synonyms and antonyms were created from WordNet and Roget's Thesaurus (Zhang et al., 2014; Ono et al., 2015); LE constraints were collected from WordNet by Vulić and Mrkšić (2018) and contain both direct and transitively obtained LE pairs. We retain the constraints for which both words exist in the trimmed (200K) FASTTEXT vocabulary, resulting in a total of 1,493,686 LE, 521,037 synonym, and 141,311 antonym pairs. We reserve 4,000 constraints (E: 2k, S: 1k, A: 1k) for validation and use the rest for training. We identify the following best hyperparameter configuration via grid search: $H = 5$, $d_h = 300$, $\psi = \tanh$, $\delta_a = 1$, $\delta_s = \delta_d = 0.5$, $\lambda_a = 2$, and $\lambda_r = 1$.

[3]For a comprehensive downstream comparison of different cross-lingual embedding models, see (Glavaš et al., 2019).

Setup	0%	10%	30%	50%	70%	90%	100%
LEAR	.174	.188	.273	.438	**.548**	**.634**	**.682**
GLEN	**.481**	**.485**	**.478**	**.474**	.506	.504	.520

Table 1: Spearman correlation for GLEN, compared with LEAR (Vulić and Mrkšić, 2018), on HyperLex, for different word coverage settings (i.e., percentages of Hyperlex words *seen* in constraints in training).

We apply a dropout (keep rate 0.5) to each hidden layer of f. We train in mini-batches of $K = 50$ constraints and learn with the Adam algorithm (Kingma and Ba, 2015): initial learning rate 10^{-4}.

3.1 Graded Lexical Entailment

We use I_{LE} to predict the strength of LE between words. We evaluate GLEN against the state-of-the-art LE-retrofitting model LEAR (Vulić and Mrkšić, 2018) on the HyperLex dataset (Vulić et al., 2017) which contains 2,616 word pairs (83% nouns, 17% verbs) judged (0-6 scale) by human annotators for the degree to which the LE relation holds. We evaluate the models in a deliberately controlled setup: we (randomly) select a subset of HyperLex words (0%, 10%, 30%, 50%, 70%, 90%, and 100%) that we allow models to "see" in the constraints, removing constraints with any other HyperLex word.[4]

Results and Discussion. The graded LE performance is shown in Table 1 for all seven setups. Graded LE results suggest that GLEN is robust and generalizes well to unseen words: the drop in performance between the 0% and 100% setups is mere 4% for GLEN (compared to a 50% drop for LEAR). Results in the 0% setting, in which GLEN improves over the distributional space by more than 30 points most clearly demonstrate its effectiveness.[5] GLEN, however, lags behind LEAR in setups where LEAR has seen 70% or more of test words. This is intuitive: LEAR specializes the vector of each particular word using only the constraints containing that word; this gives LEAR higher specialization flexibility at the expense of generalization ability. In contrast, GLEN's specialization function is affected by *all* constraints and has to work for *all* words; GLEN trades the effectiveness of LEAR's word-specific updates for seen words, for the ability to generalize over unseen words. In a sense, there is a trade-off between the ability to generalize the

[4]In the 0% setting we remove all constraints containing any HyperLex word; in the 100% we use all constraints. The full set of constraints contains 99.8% of all HyperLex words.

[5]LEAR's performance in the 0% setup corresponds to the performance of input distributional vectors.

LE-specialization over unseen words and the performance for seen words. Put differently, by learning a general specialization function – i.e., by using linguistic constraints merely as training instances – GLEN is prevented from *"overfitting"* to seen words. Evaluation settings like our 90% or 100% settings, in which GLEN is outperformed by a pure retrofitting model, are however unrealistic in view of downstream tasks: for any concrete downstream task (e.g., textual entailment or taxonomy induction), it is highly unlikely that the LE-specialization model will have seen almost all of the test words (words for which LE inference is required) in its training linguistic constraints; this is why GLEN's ability to generalize LE-specialization to unseen words (as indicated by 0%-50% settings) is particularly important.

3.2 Cross-Lingual LE Detection

Neither joint (Nguyen et al., 2017) nor retrofitting models (Vulić and Mrkšić, 2018) can predict LE across languages in a straightforward fashion. Coupled with a CL space, GLEN can seamlessly predict LE across language boundaries.

Experimental Setup. We evaluate GLEN on datasets from Upadhyay et al. (2018), encompassing two binary cross-lingual LE detection tasks: (1) HYPO task test model's ability to determine the direction of the LE relation, i.e., to discern hyponym-hypernym pairs from hypernym-hyponym pairs; (2) COHYP tasks tests whether the models are able to discern true LE pairs from cohyponyms (e.g., *car* and *boat*, cohyponyms of *vehicle*). We report results for three language pairs: English (EN) – {French (FR), Russian (RU), Arabic (AR)}. Upadhyay et al. (2018) divided each dataset into train (400-500 word pairs) and test portions (900-1000 word pairs): we use the train portions to tune the threshold t that binarizes GLEN's predictions I_{LE}.

We induce the CL embeddings (i.e., learn the projections W_g, see Section §2) by projecting AR, FR, and RU embeddings to the EN space in a supervised fashion, by finding the optimal solution to the Procrustes problem for given 5K word translation pairs (for each language pair). [6] We compare GLEN with more complex models from (Upadhyay et al., 2018): they couple two methods for inducing syntactic CL embeddings – CL-DEP (Vulić, 2017) and BI-SPARSE (Vyas and Carpuat, 2016) – with an LE scorer based on the distributional inclusion hypothesis (Geffet and Dagan, 2005).

	Model	EN-FR	EN-RU	EN-AR	Avg
HYPO	CL-DEP	.538	.602	.567	.569
	BI-SPARSE	.566	.590	.526	.561
	GLEN	**.792**	**.811**	**.816**	**.806**
COHYP	CL-DEP	.610	.562	.631	.601
	BI-SPARSE	.667	.636	.668	.657
	GLEN	**.779**	**.849**	**.821**	**.816**

Table 2: CL LE detection results (accuracy) on CL datasets (HYPO, COHYP) (Upadhyay et al., 2018).

Results. GLEN's cross-lingual LE detection performance is shown in Table 2. GLEN dramatically outperforms CL LE detection models from (Upadhyay et al., 2018), with an average edge of 24% on HYPO datasets and 16% on the COHYP datasets.[7] This accentuates GLEN's generalization ability: it robustly predicts CL LE, although trained only on EN constraints. GLEN performs better for EN-AR and EN-RU than for EN-FR: we believe this to merely be an artifact of the (rather small) test sets. We find GLEN's CL performance for more distant language pairs (EN-AR, EN-RU) especially encouraging as it holds promise of successful transfer of LE-specialization to resource-lean languages lacking external linguistic resources.

4 Conclusion

We presented GLEN, a general framework for specializing word embeddings for lexical entailment. Unlike existing LE-specialization models (Nguyen et al., 2017; Vulić and Mrkšić, 2018), GLEN learns an explicit specialization function using linguistic constraints as training examples. The learned LE-specialization function is then applied to vectors of words (1) unseen in constraints and (2) from different languages. GLEN displays robust graded LE performance and yields massive improvements over state-of-the-art in cross-lingual LE detection. We next plan to evaluate GLEN on multilingual and cross-lingual graded LE datasets (Vulić et al., 2019) and release a large multilingual repository of LE-specialized embeddings. We make GLEN (code and resources) available at: `https://github.com/codogogo/glen`.

Acknowledgments

The work of the first author was supported by the Eliteprogramm of the Baden-Württemberg Stiftung, within the scope of the AGREE grant.

[6] We automatically translated 5K most frequent EN words to AR, FR, and RU with Google Translate.

[7] All differences are statistically significant at $\alpha = 0.01$, according to the non-parametric shuffling test (Yeh, 2000)

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of ACL*, pages 789–798.

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.

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

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of ICLR*.

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*. MIT Press.

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

Goran Glavaš and Ivan Vulić. 2018. Explicit retrofitting of distributional word vectors. In *Proceedings of ACL*, pages 34–45.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. *arXiv preprint arXiv:1902.00508*.

Goran Glavaš and Simone Paolo Ponzetto. 2017. Dual tensor model for detecting asymmetric lexico-semantic relations. In *Proceedings of EMNLP*, pages 1758–1768.

Amit Gupta, Rémi Lebret, Hamza Harkous, and Karl Aberer. 2017. Taxonomy induction using hypernym subsequences. In *Proceedings of CIKM*, pages 1329–1338.

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

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of EMNLP*, pages 2979–2984.

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *Proceedings of EMNLP*, pages 2044–2048.

Diederik P. Kingma and Jimmy Ba. 2015. ADAM: A Method for Stochastic Optimization. In *Proceedings of ICLR*.

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

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*, pages 1030–1040.

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, 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.

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.

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

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Edoardo Maria Ponti, Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen. 2018. Adversarial propagation and zero-shot cross-lingual transfer of word vector specialization. In *Proceedings of EMNLP*, pages 282–293.

Marek Rei, Daniela Gerz, and Ivan Vulić. 2018. Scoring lexical entailment with a supervised directional similarity network. In *Proceedings of ACL*, pages 638–643.

Sebastian Ruder, Ryan Cotterell, Yova Kementchedjhieva, and Anders Søgaard. 2018a. A discriminative latent-variable model for bilingual lexicon induction. In *Proceedings of EMNLP*, pages 458–468.

Sebastian Ruder, Anders Søgaard, and Ivan Vulić. 2018b. A survey of cross-lingual embedding models. *arXiv preprint arXiv:1706.04902*.

Peter H Schönemann. 1966. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10.

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.

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.

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*.

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.

Shyam Upadhyay, Yogarshi Vyas, Marine Carpuat, and Dan Roth. 2018. Robust cross-lingual hypernymy detection using dependency context. In *Proceedings of NAACL*, pages 607–618.

Ivan Vulić. 2017. Cross-lingual syntactically informed distributed word representations. In *Proceedings of EACL*, volume 2, pages 408–414.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835.

Ivan Vulić and Nikola Mrkšić. 2018. Specialising word vectors for lexical entailment. In *Proceedings of NAACL-HLT*, pages 1134–1145.

Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. 2019. Multilingual and cross-lingual graded lexical entailment. In *Proceedings of ACL*, page *in print*.

Yogarshi Vyas and Marine Carpuat. 2016. Sparse bilingual word representations for cross-lingual lexical entailment. In *Proceedings of NAACL*, pages 1187–1197.

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.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, pages 1112–1122.

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.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of COLING*, pages 947–953.

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.