Retrofitting Word Vectors to Semantic Lexicons

Manaal Faruqui Jesse Dodge Sujay K. Jauhar
Chris Dyer Eduard Hovy Noah A. Smith
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
{mfaruqui,jessed,sjauhar,cdyer,ehovy,nasmith}@cs.cmu.edu

Abstract

Vector space word representations are learned from distributional information of words in large corpora. Although such statistics are semantically informative, they disregard the valuable information that is contained in semantic lexicons such as WordNet, FrameNet, and the Paraphrase Database. This paper proposes a method for refining vector space representations using relational information from semantic lexicons by encouraging linked words to have similar vector representations, and it makes no assumptions about how the input vectors were constructed. Evaluated on a battery of standard lexical semantic evaluation tasks in several languages, we obtain substantial improvements starting with a variety of word vector models. Our refinement method outperforms prior techniques for incorporating semantic lexicons into word vector training algorithms.

1 Introduction

Data-driven learning of word vectors that capture lexico-semantic information is a technique of central importance in NLP. These word vectors can in turn be used for identifying semantically related word pairs (Turney, 2006; Agirre et al., 2009) or as features in downstream text processing applications (Turian et al., 2010; Guo et al., 2014). A variety of approaches for constructing vector space embeddings of vocabularies are in use, notably including taking low rank approximations of cooccurrence statistics (Deerwester et al., 1990) and using internal representations from neural network models of word sequences (Collobert and Weston, 2008).

Because of their value as lexical semantic representations, there has been much research on improving the quality of vectors. *Semantic lexicons*, which provide type-level information about the semantics of words, typically by identifying *synonymy*, *hypernymy*, *hyponymy*, and *paraphrase* relations should be a valuable resource for improving the quality of word vectors that are trained solely on unlabeled corpora. Examples of such resources include WordNet (Miller, 1995), FrameNet (Baker et al., 1998) and the Paraphrase Database (Ganitkevitch et al., 2013).

Recent work has shown that by either changing the objective of the word vector training algorithm in neural language models (Yu and Dredze, 2014; Xu et al., 2014; Bian et al., 2014; Fried and Duh, 2014) or by relation-specific augmentation of the cooccurence matrix in spectral word vector models to incorporate semantic knowledge (Yih et al., 2012; Chang et al., 2013), the quality of word vectors can be improved. However, these methods are limited to particular methods for constructing vectors.

The contribution of this paper is a graph-based learning technique for using lexical relational resources to obtain higher quality semantic vectors, which we call "retrofitting." In contrast to previous work, retrofitting is applied as a *post-processing step* by running belief propagation on a graph constructed from lexicon-derived relational information to update word vectors (§2). This allows retrofitting to be used on pre-trained word vectors obtained using *any* vector training model. Intuitively, our method encourages the new vectors to be (i) similar to the vectors of related word types and (ii) similar to their purely distributional representations. The retrofitting process is fast, taking about 5 seconds for a graph of 100,000 words and vector length 300, and its runtime is independent of the original word vector training model.

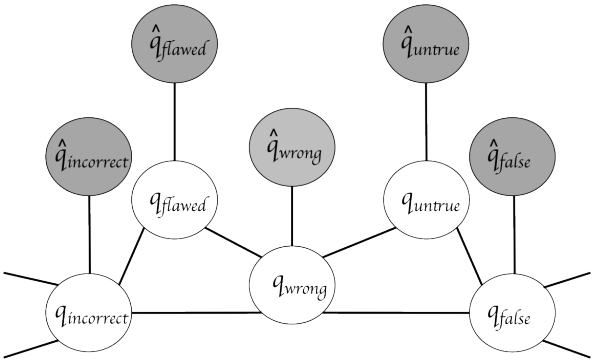

Figure 1: Word graph with edges between related words showing the observed (grey) and the inferred (white) word vector representations.

Experimentally, we show that our method works well with different state-of-the-art word vector models, using different kinds of semantic lexicons and gives substantial improvements on a variety of benchmarks, while beating the current state-of-the-art approaches for incorporating semantic information in vector training and trivially extends to multiple languages. We show that retrofitting gives consistent improvement in performance on evaluation benchmarks with different word vector lengths and show a qualitative visualization of the effect of retrofitting on word vector quality. The retrofitting tool is available at: `https://github.com/mfaruqui/retrofitting`.

2 Retrofitting with Semantic Lexicons

Let $V = \{w_1, \ldots, w_n\}$ be a **vocabulary**, i.e, the set of word types, and Ω be an **ontology** that encodes semantic relations between words in V. We represent Ω as an undirected graph (V, E) with one vertex for each word type and edges $(w_i, w_j) \in E \subseteq V \times V$ indicating a semantic relationship of interest. These relations differ for different semantic lexicons and are described later (§4).

The matrix $\hat{Q}$ will be the collection of vector representations $\hat{q}_i \in \mathbb{R}^d$, for each $w_i \in V$, learned using a standard data-driven technique, where d is the length of the word vectors. Our objective is to learn the matrix $Q = (q_1, \ldots, q_n)$ such that the columns are both close (under a distance metric) to their counterparts in $\hat{Q}$ *and* to adjacent vertices in Ω. Figure 1 shows a small word graph with such edge connections; white nodes are labeled with the Q vectors to be retrofitted (and correspond to V_Ω); shaded nodes are labeled with the corresponding vectors in $\hat{Q}$, which are observed. The graph can be interpreted as a Markov random field (Kindermann and Snell, 1980).

The distance between a pair of vectors is defined to be the Euclidean distance. Since we want the inferred word vector to be close to the observed value $\hat{q}_i$ and close to its neighbors $q_j, \forall j$ such that $(i, j) \in E$, the objective to be minimized becomes:

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

where α and β values control the relative strengths of associations (more details in §6.1).

In this case, we first train the word vectors independent of the information in the semantic lexicons and then retrofit them. Ψ is convex in Q and its solution can be found by solving a system of linear equations. To do so, we use an efficient iterative updating method (Bengio et al., 2006; Subramanya et al., 2010; Das and Petrov, 2011; Das and Smith, 2011). The vectors in Q are initialized to be equal to the vectors in $\hat{Q}$. We take the first derivative of Ψ with respect to one q_i vector, and by equating it to zero arrive at the following online update:

$$q_i = \frac{\sum_{j:(i,j)\in E} \beta_{ij} q_j + \alpha_i \hat{q}_i}{\sum_{j:(i,j)\in E} \beta_{ij} + \alpha_i} \quad (1)$$

In practice, running this procedure for 10 iterations converges to changes in Euclidean distance of adjacent vertices of less than 10^{-2}. The retrofitting approach described above is modular; it can be applied to word vector representations obtained from any model as the updates in Eq. 1 are agnostic to the original vector training model objective.

Semantic Lexicons during Learning. Our proposed approach is reminiscent of recent work on improving word vectors using lexical resources (Yu and Dredze, 2014; Bian et al., 2014; Xu et al., 2014) which alters the learning objective of the original vector training model with a prior (or a regularizer) that encourages semantically related vectors (in Ω) to be close together, except that our technique is applied as a second stage of learning. We describe the

prior approach here since it will serve as a baseline. Here semantic lexicons play the role of a prior on Q which we define as follows:

$$p(Q) \propto \exp\left(-\gamma \sum_{i=1}^{n} \sum_{j:(i,j)\in E} \beta_{ij} \|q_i - q_j\|^2\right) \quad (2)$$

Here, γ is a hyperparameter that controls the strength of the prior. As in the retrofitting objective, this prior on the word vector parameters forces words connected in the lexicon to have close vector representations as did $\Psi(Q)$ (with the role of $\hat{Q}$ being played by cross entropy of the empirical distribution).

This prior can be incorporated during learning through maximum a posteriori (MAP) estimation. Since there is no closed form solution of the estimate, we consider two iterative procedures. In the first, we use the sum of gradients of the log-likelihood (given by the extant vector learning model) and the log-prior (from Eq. 2), with respect to Q for learning. Since computing the gradient of Eq. 2 has linear runtime in the vocabulary size n, we use lazy updates (Carpenter, 2008) for every k words during training. We call this the **lazy** method of MAP. The second technique applies stochastic gradient ascent to the log-likelihood, and after every k words applies the update in Eq. 1. We call this the **periodic** method. We later experimentally compare these methods against retrofitting (§6.2).

3 Word Vector Representations

We now describe the various publicly available pretrained English word vectors on which we will test the applicability of the retrofitting model. These vectors have been chosen to have a balanced mix between large and small amounts of unlabeled text as well as between neural and spectral methods of training word vectors.

Glove Vectors. Global vectors for word representations (Pennington et al., 2014) are trained on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations show interesting linear substructures of the word vector space. These vectors were trained on 6 billion words from Wikipedia and English Gigaword and are of length 300.[1]

Lexicon	Words	Edges
PPDB	102,902	374,555
WordNet$_{syn}$	148,730	304,856
WordNet$_{all}$	148,730	934,705
FrameNet	10,822	417,456

Table 1: Approximate size of the graphs obtained from different lexicons.

Skip-Gram Vectors (SG). The `word2vec` tool (Mikolov et al., 2013a) is fast and currently in wide use. In this model, each word's Huffman code is used as an input to a log-linear classifier with a continuous projection layer and words within a given context window are predicted. The available vectors are trained on 100 billion words of Google news dataset and are of length 300.[2]

Global Context Vectors (GC). These vectors are learned using a recursive neural network that incorporates both local and global (document-level) context features (Huang et al., 2012). These vectors were trained on the first 1 billion words of English Wikipedia and are of length 50.[3]

Multilingual Vectors (Multi). Faruqui and Dyer (2014) learned vectors by first performing SVD on text in different languages, then applying canonical correlation analysis (CCA) on pairs of vectors for words that align in parallel corpora. The monolingual vectors were trained on WMT-2011 news corpus for English, French, German and Spanish. We use the Enligsh word vectors projected in the common English–German space. The monolingual English WMT corpus had 360 million words and the trained vectors are of length 512.[4]

4 Semantic Lexicons

We use three different semantic lexicons to evaluate their utility in improving the word vectors. We include both manually and automatically created lexicons. Table 1 shows the size of the graphs obtained

[1] `http://www-nlp.stanford.edu/projects/glove/`

[2] `https://code.google.com/p/word2vec`

[3] `http://nlp.stanford.edu/~socherr/ACL2012_wordVectorsTextFile.zip`

[4] `http://cs.cmu.edu/~mfaruqui/soft.html`

from these lexicons.

PPDB. The paraphrase database (Ganitkevitch et al., 2013) is a semantic lexicon containing more than 220 million paraphrase pairs of English.[5] Of these, 8 million are lexical (single word to single word) paraphrases. The key intuition behind the acquisition of its lexical paraphrases is that two words in one language that align, in parallel text, to the same word in a different language, should be synonymous. For example, if the words *jailed* and *imprisoned* are translated as the same word in another language, it may be reasonable to assume they have the same meaning. In our experiments, we instantiate an edge in E for each lexical paraphrase in PPDB. The lexical paraphrase dataset comes in different sizes ranging from S to XXXL, in decreasing order of paraphrasing confidence and increasing order of size. We chose XL for our experiments. We want to give higher edge weights (α_i) connecting the retrofitted word vectors (q) to the purely distributional word vectors ($\hat{q}$) than to edges connecting the retrofitted vectors to each other (β_{ij}), so all α_i are set to 1 and β_{ij} to be $\text{degree}(i)^{-1}$ (with i being the node the update is being applied to).[6]

WordNet. WordNet (Miller, 1995) is a large human-constructed semantic lexicon of English words. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between synsets. This database is structured in a graph particularly suitable for our task because it explicitly relates concepts with semantically aligned relations such as hypernyms and hyponyms. For example, the word *dog* is a synonym of *canine*, a hypernym of *puppy* and a hyponym of *animal*. We perform two different experiments with WordNet: (1) connecting a word only to synonyms, and (2) connecting a word to synonyms, hypernyms and hyponyms. We refer to these two graphs as WN_{syn} and WN_{all}, respectively. In both settings, all α_i are set to 1 and β_{ij} to be $\text{degree}(i)^{-1}$.

[5] http://www.cis.upenn.edu/~ccb/ppdb

[6] In principle, these hyperparameters can be tuned to optimize performance on a particular task, which we leave for future work.

FrameNet. FrameNet (Baker et al., 1998; Fillmore et al., 2003) is a rich linguistic resource containing information about lexical and predicate-argument semantics in English. Frames can be realized on the surface by many different word types, which suggests that the word types evoking the same frame should be semantically related. For example, the frame Cause_change_of_position_on_a_scale is associated with *push*, *raise*, and *growth* (among many others). In our use of FrameNet, two words that group together with any frame are given an edge in E. We refer to this graph as FN. All α_i are set to 1 and β_{ij} to be $\text{degree}(i)^{-1}$.

5 Evaluation Benchmarks

We evaluate the quality of our word vector representations on tasks that test how well they capture both semantic and syntactic aspects of the representations along with an extrinsic sentiment analysis task.

Word Similarity. We evaluate our word representations on a variety of different benchmarks that have been widely used to measure word similarity. The first one is the **WS-353** dataset (Finkelstein et al., 2001) containing 353 pairs of English words that have been assigned similarity ratings by humans. The second benchmark is the **RG-65** (Rubenstein and Goodenough, 1965) dataset that contain 65 pairs of nouns. Since the commonly used word similarity datasets contain a small number of word pairs we also use the **MEN** dataset (Bruni et al., 2012) of 3,000 word pairs sampled from words that occur at least 700 times in a large web corpus. We calculate cosine similarity between the vectors of two words forming a test item, and report Spearman's rank correlation coefficient (Myers and Well, 1995) between the rankings produced by our model against the human rankings.

Syntactic Relations (SYN-REL). Mikolov et al. (2013b) present a syntactic relation dataset composed of analogous word pairs. It contains pairs of tuples of word relations that follow a common syntactic relation. For example, given *walking* and *walked*, the words are differently inflected forms of the same verb. There are nine different kinds of relations and overall there are 10,675 syntactic pairs of word tuples. The task is to find a word *d* that best

fits the following relationship: "*a* is to *b* as *c* is to *d*," given a, b, and c. We use the vector offset method (Mikolov et al., 2013a; Levy and Goldberg, 2014), computing $q = q_a - q_b + q_c$ and returning the vector from Q which has the highest cosine similarity to q.

Synonym Selection (TOEFL). The TOEFL synonym selection task is to select the semantically closest word to a target from a list of four candidates (Landauer and Dumais, 1997). The dataset contains 80 such questions. An example is "*rug* $\rightarrow$ {*sofa*, *ottoman*, *carpet*, *hallway*}", with *carpet* being the most synonym-like candidate to the target.

Sentiment Analysis (SA). Socher et al. (2013) created a treebank containing sentences annotated with fine-grained sentiment labels on phrases and sentences from movie review excerpts. The coarse-grained treebank of positive and negative classes has been split into training, development, and test datasets containing 6,920, 872, and 1,821 sentences, respectively. We train an ℓ_2-regularized logistic regression classifier on the average of the word vectors of a given sentence to predict the coarse-grained sentiment tag at the sentence level, and report the test-set accuracy of the classifier.

6 Experiments

We first show experiments measuring improvements from the retrofitting method (§6.1), followed by comparisons to using lexicons during MAP learning (§6.2) and other published methods (§6.3). We then test how well retrofitting generalizes to other languages (§6.4).

6.1 Retrofitting

We use Eq. 1 to retrofit word vectors (§3) using graphs derived from semantic lexicons (§4).

Results. Table 2 shows the absolute changes in performance on different tasks (as columns) with different semantic lexicons (as rows). All of the lexicons offer high improvements on the word similarity tasks (the first three columns). On the TOEFL task, we observe large improvements of the order of 10 absolute points in accuracy for all lexicons except for FrameNet. FrameNet's performance is weaker, in some cases leading to worse performance (e.g., with Glove and SG vectors). For the extrinsic sentiment analysis task, we observe improvements using all the lexicons and gain 1.4% (absolute) in accuracy for the Multi vectors over the baseline. This increase is statistically significant ($p < 0.01$, McNemar).

We observe improvements over Glove and SG vectors, which were trained on billions of tokens on all tasks except for SYN-REL. For stronger baselines (Glove and Multi) we observe smaller improvements as compared to lower baseline scores (SG and GC). We believe that FrameNet does not perform as well as the other lexicons because its frames group words based on very abstract concepts; often words with seemingly distantly related meanings (e.g., *push* and *growth*) can evoke the same frame. Interestingly, we almost never improve on the SYN-REL task, especially with higher baselines, this can be attributed to the fact that SYN-REL is inherently a syntactic task and during retrofitting we are incorporating additional semantic information in the vectors. In summary, we find that PPDB gives the best improvement maximum number of times aggreagted over different vetor types, closely followed by WN$_{all}$, and retrofitting gives gains across tasks and vectors. An ensemble lexicon, in which the graph is the union of the WN$_{all}$ and PPDB lexicons, on average performed slightly worse than PPDB; we omit those results here for brevity.

6.2 Semantic Lexicons during Learning

To incorporate lexicon information during training, and compare its performance against retrofitting, we train log-bilinear (LBL) vectors (Mnih and Teh, 2012). These vectors are trained to optimize the log-likelihood of a language model which predicts a word token w's vector given the set of words in its context (h), also represented as vectors:

$$p(w \mid h; Q) \propto \exp\left(\sum_{i \in h} q_i^\top q_j + b_j\right) \qquad (3)$$

We optimize the above likelihood combined with the prior defined in Eq. 2 using the lazy and periodic techniques described in §2. Since it is costly to compute the partition function over the whole vocabulary, we use *noise constrastive estimation* (NCE) to estimate the parameters of the model (Mnih and Teh, 2012) using AdaGrad (Duchi et al., 2010) with a learning rate of 0.05.

Lexicon	MEN-3k	RG-65	WS-353	TOEFL	SYN-REL	SA
Glove	73.7	76.7	60.5	89.7	67.0	79.6
+PPDB	1.4	2.9	–1.2	**5.1**	–0.4	**1.6**
+WN$_{syn}$	0.0	2.7	0.5	**5.1**	–12.4	0.7
+WN$_{all}$	**2.2**	**7.5**	**0.7**	2.6	–8.4	0.5
+FN	–3.6	–1.0	–5.3	2.6	–7.0	0.0
SG	67.8	72.8	65.6	85.3	73.9	81.2
+PPDB	**5.4**	3.5	**4.4**	**10.7**	–2.3	**0.9**
+WN$_{syn}$	0.7	3.9	0.0	9.3	–13.6	0.7
+WN$_{all}$	2.5	**5.0**	1.9	9.3	–10.7	–0.3
+FN	–3.2	2.6	–4.9	1.3	–7.3	0.5
GC	31.3	62.8	62.3	60.8	10.9	67.8
+PPDB	**7.0**	6.1	2.0	**13.1**	**5.3**	**1.1**
+WN$_{syn}$	3.6	6.4	0.6	7.3	–1.7	0.0
+WN$_{all}$	6.7	**10.2**	**2.3**	4.4	–0.6	0.2
+FN	1.8	4.0	0.0	4.4	–0.6	0.2
Multi	75.8	75.5	68.1	84.0	45.5	81.0
+PPDB	**3.8**	4.0	**6.0**	**12.0**	**4.3**	0.6
+WN$_{syn}$	1.2	0.2	2.2	6.6	–12.3	**1.4**
+WN$_{all}$	2.9	**8.5**	4.3	6.6	–10.6	**1.4**
+FN	1.8	4.0	0.0	4.4	–0.6	0.2

Table 2: Absolute performance changes with retrofitting. Spearman's correlation (3 left columns) and accuracy (3 right columns) on different tasks. Higher scores are always better. Bold indicates greatest improvement for a vector type.

Method	k, γ	MEN-3k	RG-65	WS-353	TOEFL	SYN-REL	SA
LBL (Baseline)	$k = \infty, \gamma = 0$	58.0	42.7	53.6	66.7	31.5	72.5
LBL + Lazy	$\gamma = 1$	–0.4	4.2	0.6	–0.1	0.6	1.2
	$\gamma = 0.1$	0.7	8.1	0.4	–1.4	0.7	0.8
	$\gamma = 0.01$	0.7	9.5	1.7	2.6	1.9	0.4
LBL + Periodic	$k = 100$M	3.8	18.4	3.6	12.0	4.8	1.3
	$k = 50$M	3.4	**19.5**	4.4	18.6	0.6	**1.9**
	$k = 25$M	0.5	18.1	2.7	**21.3**	–3.7	0.8
LBL + Retrofitting	–	**5.7**	15.6	**5.5**	18.6	**14.7**	0.9

Table 3: Absolute performance changes for including PPDB information while training LBL vectors. Spearman's correlation (3 left columns) and accuracy (3 right columns) on different tasks. Bold indicates greatest improvement.

We train vectors of length 100 on the WMT-2011 news corpus, which contains 360 million words, and use PPDB as the semantic lexicon as it performed reasonably well in the retrofitting experiments (§6.1). For the lazy method we update with respect to the prior every k = 100,000 words[7] and test for different values of prior strength $\gamma \in \{1, 0.1, 0.01\}$. For the periodic method, we update the word vectors using Eq. 1 every $k \in \{25, 50, 100\}$ million words.

Results. See Table 3. For lazy, $\gamma = 0.01$ performs best, but the method is in most cases not highly sensitive to γ's value. For **periodic**, which overall leads to greater improvements over the baseline than **lazy**, $k = 50$M performs best, although all other values of k also outperform the the baseline. Retrofitting, which can be applied to any word vectors, regardless of how they are trained, is competitive and sometimes better.

[7] k = 10,000 or 50,000 yielded similar results.

Corpus	Vector Training	MEN-3k	RG-65	WS-353	TOEFL	SYN-REL	SA
WMT-11	CBOW	55.2	44.8	54.7	73.3	40.8	74.1
	Yu and Dredze (2014)	50.1	47.1	53.7	61.3	29.9	71.5
	CBOW + Retrofitting	**60.5**	**57.7**	**58.4**	**81.3**	**52.5**	**75.7**
Wikipedia	SG	**76.1**	66.7	**68.6**	72.0	40.3	73.1
	Xu et al. (2014)	–	–	68.3	–	44.4	–
	SG + Retrofitting	65.7	**73.9**	67.5	**86.0**	**49.9**	**74.6**

Table 4: Comparison of retrofitting for semantic enrichment against Yu and Dredze (2014), Xu et al. (2014). Spearman's correlation (3 left columns) and accuracy (3 right columns) on different tasks.

6.3 Comparisons to Prior Work

Two previous models (Yu and Dredze, 2014; Xu et al., 2014) have shown that the quality of word vectors obtained using `word2vec` tool can be improved by using semantic knowledge from lexicons. Both these models use constraints among words as a regularization term on the training objective during training, and their methods can only be applied for improving the quality of SG and CBOW vectors produced by the `word2vec` tool. We compared the quality of our vectors against each of these.

Yu and Dredze (2014). We train word vectors using their joint model training code[8] while using exactly the same training settings as specified in their best model: CBOW, vector length 100 and PPDB for enrichment. The results are shown in the top half of Table 4 where our model consistently outperforms the baseline and their model.

Xu et al. (2014). This model extracts categorical and relational knowledge among words from Freebase[9] and uses it as a constraint while training. Unfortunately, neither their word embeddings nor model training code is publicly available, so we train the SG model by using exactly the same settings as described in their system (vector length 300) and on the same corpus: monolingual English Wikipedia text.[10] We compare the performance of our retrofitting vectors on the SYN-REL and WS-353 task against the best model[11] reported in their paper. As shown in the lower half of Table 4, our model outperforms their model by an absolute 5.5 points absolute on the SYN-REL task, but a slightly inferior score on the WS-353 task.

6.4 Multilingual Evaluation

We tested our method on three additional languages: German, French, and Spanish. We used the Universal WordNet (de Melo and Weikum, 2009), an automatically constructed multilingual lexical knowledge base based on WordNet.[12] It contains words connected via different lexical relations to other words both within and across languages. We construct separate graphs for different languages (i.e., only linking words to other words in the same language) and apply retrofitting to each. Since not many word similarity evaluation benchmarks are available for languages other than English, we tested our baseline and improved vectors on one benchmark per language.

We used RG-65 (Gurevych, 2005), RG-65 (Joubarne and Inkpen, 2011) and MC-30 (Hassan and Mihalcea, 2009) for German, French and Spanish, respectively.[13] We trained SG vectors for each language of length 300 on a corpus of 1 billion tokens, each extracted from Wikipedia, and evaluate them on word similarity on the benchmarks before and after retrofitting. Table 5 shows that we obtain high improvements which strongly indicates that our method generalizes across these languages.

7 Further Analysis

Retrofitting vs. vector length. With more dimensions, word vectors might be able to capture higher orders of semantic information and retrofitting might be less helpful. We train SG vec-

[8] `https://github.com/Gorov/JointRCM`

[9] `https://www.freebase.com`

[10] `http://mattmahoney.net/dc/enwik9.zip`

[11] Their best model is named "RC-NET" in their paper.

[12] `http://www.mpi-inf.mpg.de/yago-naga/uwn`

[13] These benchmarks were created by translating the corresponding English benchmarks word by word manually.

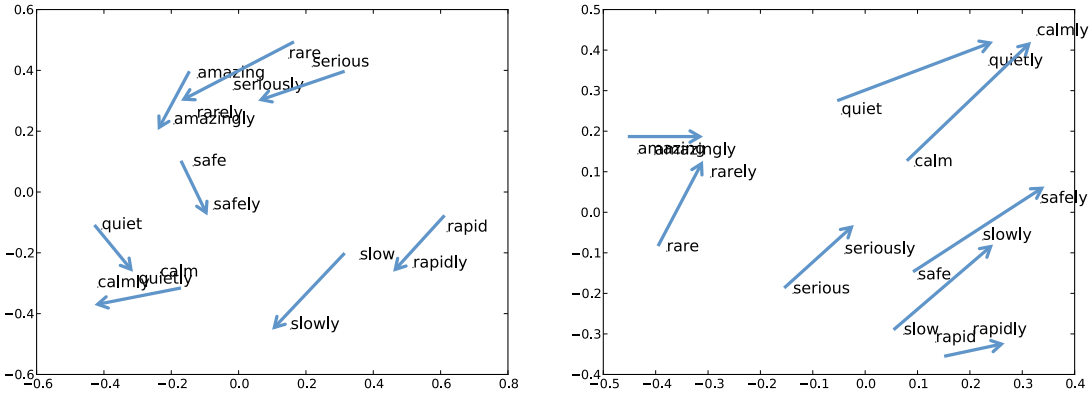

Figure 3: Two-dimensional PCA projections of 100-dimensional SG vector pairs holding the "adjective to adverb" relation, before (left) and after (right) retrofitting.

Language	Task	SG	Retrofitted SG
German	RG-65	53.4	**60.3**
French	RG-65	46.7	**60.6**
Spanish	MC-30	54.0	**59.1**

Table 5: Spearman's correlation for word similarity evaluation using the using original and retrofitted SG vectors.

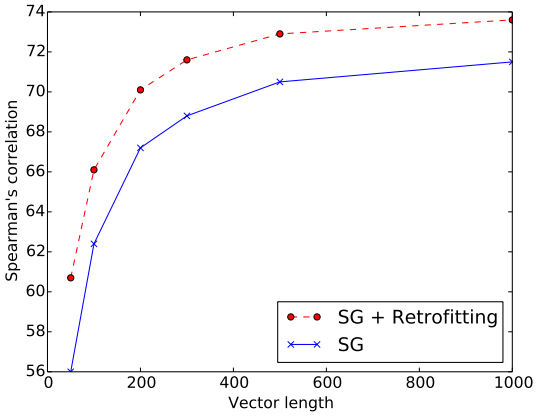

Figure 2: Spearman's correlation on the MEN word similarity task, before and after retrofitting.

tors on 1 billion English tokens for vector lengths ranging from 50 to 1,000 and evaluate on the MEN word similarity task. We retrofit these vectors to PPDB (§4) and evaluate those on the same task. Figure 2 shows consistent improvement in vector quality across different vector lengths.

Visualization. We randomly select eight word pairs that have the "adjective to adverb" relation from the SYN-REL task (§5). We then take a two-dimensional PCA projection of the 100-dimensional SG word vectors and plot them in $\mathbb{R}^2$. In Figure 3 we plot these projections before (left) and after (right) retrofitting. It can be seen that in the first case the direction of the analogy vectors is not consistent, but after retrofitting all the analogy vectors are aligned in the same direction.

8 Related Work

The use of lexical semantic information in training word vectors has been limited. Recently, word similarity knowledge (Yu and Dredze, 2014; Fried and Duh, 2014) and word relational knowledge (Xu et al., 2014; Bian et al., 2014) have been used to improve the `word2vec` embeddings in a joint training model similar to our regularization approach. In latent semantic analysis, the word cooccurrence matrix can be constructed to incorporate relational information like antonym specific polarity induction (Yih et al., 2012) and multi-relational latent semantic analysis (Chang et al., 2013).

The approach we propose is conceptually similar to previous work that uses graph structures to propagate information among semantic concepts (Zhu, 2005; Culp and Michailidis, 2008). Graph-based belief propagation has also been used to induce POS tags (Subramanya et al., 2010; Das and Petrov, 2011) and semantic frame associations (Das and Smith, 2011). In those efforts, labels for unknown words were inferred using a method similar to ours. Broadly, graph-based semi-supervised learning (Zhu, 2005; Talukdar and Pereira, 2010) has been applied to machine translation (Alexandrescu

and Kirchhoff, 2009), unsupervised semantic role induction (Lang and Lapata, 2011), semantic document modeling (Schuhmacher and Ponzetto, 2014), language generation (Krahmer et al., 2003) and sentiment analysis (Goldberg and Zhu, 2006).

9 Conclusion

We have proposed a simple and effective method named **retrofitting** to improve word vectors using word relation knowledge found in semantic lexicons. Retrofitting is used as a post-processing step to improve vector quality and is more modular than other approaches that use semantic information while training. It can be applied to vectors obtained from any word vector training method. Our experiments explored the method's performance across tasks, semantic lexicons, and languages and showed that it outperforms existing alternatives. The retrofitting tool is available at: `https://github.com/mfaruqui/retrofitting`.

Acknowledgements

This research was supported in part by the National Science Foundation under grants IIS-1143703, IIS-1147810, and IIS-1251131; by IARPA via Department of Interior National Business Center (DoI/NBC) contract number D12PC00337; and by DARPA under grant FA87501220342. Part of the computational work was carried out on resources provided by the Pittsburgh Supercomputing Center. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: the views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, DARPA, or the U.S. Government.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of NAACL*.

Andrei Alexandrescu and Katrin Kirchhoff. 2009. Graph-based learning for statistical machine translation. In *Proceedings of NAACL*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of ACL*.

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 2006. Label propagation and quadratic criterion. In *Semi-Supervised Learning*.

Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Knowledge-powered deep learning for word embedding. In *Machine Learning and Knowledge Discovery in Databases*.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of ACL*.

Bob Carpenter. 2008. Lazy sparse stochastic gradient descent for regularized multinomial logistic regression. Technical Report Alias-i Inc.

Kai-Wei Chang, Wen-tau Yih, and Christopher Meek. 2013. Multi-relational latent semantic analysis. In *Proceedings of EMNLP*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of ICML*.

Mark Culp and George Michailidis. 2008. Graph-based semisupervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proc. of ACL*.

Dipanjan Das and Noah A. Smith. 2011. Semi-supervised frame-semantic parsing for unknown predicates. In *Proc. of ACL*.

Gerard de Melo and Gerhard Weikum. 2009. Towards a universal wordnet by learning from combined evidence. In *Proceedings of CIKM*.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*.

John Duchi, Elad Hazan, and Yoram Singer. 2010. Adaptive subgradient methods for online learning and stochastic optimization. Technical Report UCB/EECS-2010-24, Mar.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL*.

Charles Fillmore, Christopher Johnson, and Miriam Petruck. 2003. *International Journal of Lexicography*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *Proceedings of WWW*.

Daniel Fried and Kevin Duh. 2014. Incorporating both distributional and relational semantics in word representations. *arXiv preprint arXiv:1412.4369*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL*.

Andrew B. Goldberg and Xiaojin Zhu. 2006. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. TextGraphs-1.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Revisiting embedding features for simple semi-supervised learning. In *Proceedings of EMNLP*.

Iryna Gurevych. 2005. Using the structure of a conceptual network in computing semantic relatedness. In *Proceedings of IJCNLP*.

Samer Hassan and Rada Mihalcea. 2009. Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proc. of EMNLP*.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of ACL*.

Colette Joubarne and Diana Inkpen. 2011. Comparison of semantic similarity for different languages using the google n-gram corpus and second- order co-occurrence measures. In *Proceedings of CAAI*.

Ross Kindermann and J. L. Snell. 1980. *Markov Random Fields and Their Applications*. AMS.

Emiel Krahmer, Sebastian van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Comput. Linguist.*

Thomas K Landauer and Susan T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*.

Joel Lang and Mirella Lapata. 2011. Unsupervised semantic role induction with graph partitioning. In *Proceedings of EMNLP*.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of CoNLL*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of ICML*.

Jerome L. Myers and Arnold D. Well. 1995. *Research Design & Statistical Analysis*. Routledge.

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

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October.

Michael Schuhmacher and Simone Paolo Ponzetto. 2014. Knowledge-based graph document modeling. In *Proceedings of WSDM*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

Amarnag Subramanya, Slav Petrov, and Fernando Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of EMNLP*.

Partha Pratim Talukdar and Fernando Pereira. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of ACL*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proc. of ACL*.

Peter D. Turney. 2006. Similarity of semantic relations. *Comput. Linguist.*, 32(3):379–416, September.

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*.

Wen-tau Yih, Geoffrey Zweig, and John C. Platt. 2012. Polarity inducing latent semantic analysis. In *Proceedings of EMNLP*.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *ACL*.

Xiaojin Zhu. 2005. *Semi-supervised Learning with Graphs*. Ph.D. thesis, Pittsburgh, PA, USA. AAI3179046.