Retrofitting Word Vectors to Semantic Lexicons

Abstract

Recent advances in vector space word representation learning have only used evidence from co-occurrence statistics in monolingual and multilingual text corpora. All such techniques are consequently unaware of the crucial information present in handcrafted and automatically produced semantic lexicons that is not clearly evident from text. In this paper, we propose a novel approach that combines distributional information with additional knowledge from such semantic lexicons to yield better quality word representations than in isolation. We perform belief propagation on a graph constructed using available semantic lexicons to enforce connected words to have similar representations and show that our technique is applicable to different word vectors models across different languages and gives improvement in performance on a variety of evaluation tasks.

1 Introduction

Data-driven learning of word vectors that capture lexico-semantic properties is a technique of central importance in natural language processing. These word vectors can in turn be used for identifying semantically close word pairs (Turney, 2006; Agirre et al., 2009) or as features in downstream applications like named entity recognition (Turian et al., 2010). It is possible to construct high-quality word vectors using cooccurrence statistics from a large corpus of text (Deerwester et al., 1990), or using internal representations from neural network models of word sequences (Collobert and Weston, 2008). Variants of using cooccurrence statistics from monolingual corpora by combining extra information from multilingual context (Zou et al., 2013; Hermann and Blunsom, 2014; Faruqui and Dyer, 2014) and dependency-based context (Padó and Lapata, 2007) have been shown to perform better than their original counterparts.

In a similar spirit, we propose that word vector representations can be further improved by looking at the neighboring words in a semantic lexicon. Semantic lexicons like the WordNet (Miller, 1995), FrameNet (Baker et al., 1998) or the Paraphrase database (Ganitkevitch et al., 2013) contain words that are connected to other words in the same language through specific relations like *synonymy*, *hypernymy*, *hyponymy*, *paraphrases* etc. . These relations in turn determine the semantic association between the words. In particular, the existence of a *synonymy* relation between two words is a strong indication for them to have similar vector representations. For example, in WordNet the words *scream* and *yell* appear as synonyms and hence we posit that there word vector representations should be close. This is an important source of information that has not been systematically exploited in creating word vectors so far.

We present a graph-based learning framework which uses the semantic lexicons to create a word graph (§2) where semantically associated words are connected to each other. We then perform belief propagation on the graph to enforce similar words to have similar word vector representations. Our method of retrofittingword vectors to semantic lexicons can be used both as a post-processing step (§2.1) or during training the word vectors (§2.2). We show that our method works well with different types of word vector models (§3) while using different kinds of semantic lexicons (§4) and gives substantial improvements on a variety of vector evaluations tasks (§5) across multiple languages (§6.3). We also show that our method is capable of easily inducing sense-specific vectors for words with multiple word senses (§7).

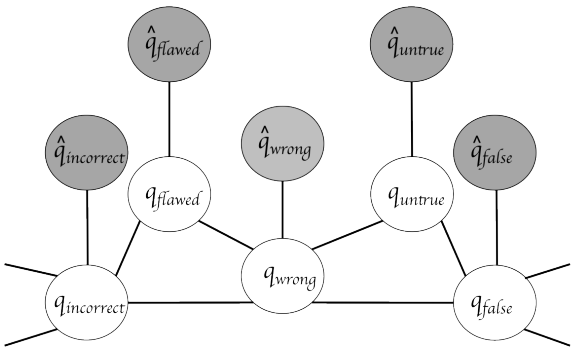

Figure 1: Word graph with edges between related words showing the observed and the inferred word vector representations.

2 Retrofitting Framework

Let $W = \{w_1, ..., w_n\}$ be the set of word types and Ω be an ontology that encodes a set of semantic relations between words in W. Specifically, we define $\Omega = (V_\Omega, E_\Omega)$ to be an undirected graph with vertices $V_\Omega = \{v_i | \ \forall \ w_i \in W\}$ and edges $E_\Omega = \{e_{ij}\}$ for every pair of words (w_i, w_j) that are semantically linked according to different semantic relations. These relations differ for different semantic lexicons as described later in §4. Since majority of the existing word representation models ignore word-senses, we consider $\forall i, w_i$ to represent undisambiguated word-types, however our model can be augmented to induce representations for sense-specific vectors as well (§7).

Then, given a set of word vectors $\hat{Q} = \{\hat{q}_i | \ \forall \ w_i \in W\}$ that have either been learned from any of the currently available methods (described later in §3) or are currently being learnt, our objective is to learn a set of vectors $Q = \{q_i | \ \forall \ w_i \in W\}$ that are consistent with both $\hat{Q}$ and neighboring nodes in Ω by a notion of distance metric between vectors. Figure 1 shows a small word graph with such edge connections. The sets of nodes containing vectors Q and vectors $\hat{Q}$ can be seen as the unobserved and observed nodes in a markov random field (Kindermann and Snell, 1980). We initialize the vectors in Q to be equal to the vectors in $\hat{Q}$ and then perform inference to obtain the unobserved (and retrofitted) word representations Q.

We define the distances between any two nodes as the euclidean distance[1] between their word vectors, thus forcing neighboring nodes to have similar vector representations. Since, we want the inferred word vector to be associated both with its observed value $\hat{q}_i$ and its neighbors $q_j, \forall ij \in E_\Omega$, the objective becomes:

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

where, β_{ij} is the weight of the edges (details in §6.1) connecting i, j and α_i is a parameter to control the degree of deviation of the word vector from its observed value. We now look at two different ways of using this objective for retrofitting the vectors to semantic lexicons: as a post-processing step after training the word vectors and during training the word vectors.

2.1 Retrofitting by Post-processing

In this case, we first train the word vectors independent of the information in the semantic lexicons and later retrofit them as a post-processing step. Equation 1 is a convex optimization problem whose solution is solving a system of linear equations involving inversion of a large matrix which is $O(n^3)$, thus instead we use an iterative update schema for optimization (Bengio et al., 2006; Subramanya et al., 2010; Das and Petrov, 2011; Das and Smith, 2011). We take the gradient of equation 1 with respect to the parameters to be optimized, namely: $\forall i, q_i$ and equate it to zero to get the following update for q_i:

$$q_i = \frac{\sum_{j,ij \in E} \beta_{ij} q_j + \alpha \hat{q}_i}{\sum_{j,ij \in E} \beta_{ij} + \alpha} \tag{2}$$

In practice running this procedure for 10 iterations leads to convergence. This is a simple post-processing model that can work with any kind of word vector representations.

2.2 Bayesian Retrofitting

Training neural language models include performing maximum likelihood estimation (MLE) on the word vector parameters (Collobert and Weston, 2008; Mnih and Teh, 2012; Mikolov et al., 2013a). These models maximize $p(w|h; Q)$, the probability of observing a word w given a sequence of contextual words h and the word vectors Q. In addition to this, we formulate the information from the semantic lexicons as a prior on the parameters Q

[1] Theoretically, using negative cosine similarity as distance yields the same update equation for vectors of unit length

and instead perform a maximum a posteriori estimation (MAP). The prior on Q can be defined as:

$$p(Q) \propto \exp\left(-\gamma \sum_{i=1}^{n} \sum_{ij \in E_\Omega} \beta_{ij} \|q_i - q_j\|^2\right) \quad (3)$$

where γ is a hyper-parameter that controls the strength of the prior. This prior on the word vector parameters forces words connected in the lexicon to have close vector representations similar to equation 1.

2.3 Optimization

The optimization of the objective during post-processing (§2.1) is a series of iterative updates of the parameters Q. For the bayesian framework (§2.2) we use two different techniques for performing MAP.

In the first technique, we take the derivative of the log prior (equ. 3) with respect to the parameters Q and add it to the gradient of the posterior, thus optimizing them jointly using adaptive gradient descent (Duchi et al., 2010). However, since computing gradient of equ. 3 is linear in the vocabulary size n, we use lazy updates (Carpenter, 2008) every k words during training i.e, if the gradient update for one training instance is g, the lazy gradient would be kg and would be computed after every k training instances. We call this the **XYZ** method of retrofitting during training. The second technique is the stochastic gradient descent method of optimization while training. While optimizing the posterior $p(w|h;Q)$ we re-estimate the word vectors using equ. 2 after we visit every k words during training. We call this the **ABC** method of retrofitting during training.

3 Word Vector Representations

We test our retrofitting model on a number of different models of word vector representations described below. We train some models on our data (§3.1,§3.2,§3.5) and use pre-trained vectors from other models (§3.3,§3.4).

3.1 Latent Semantic Analysis (LSA)

We perform latent semantic analysis (Deerwester et al., 1990) on a word-word co-occurrence matrix. We construct a word co-occurrence frequency matrix for a given training corpus where each row $\boldsymbol{w}$, represents one word in the corpus and every column $\boldsymbol{c}$, is the context feature in which the word is observed. In our case, every column is a word which occurs in a given window length around the target word. For scalability reasons, we only select words with frequency greater than 10 as features. We also remove the top 100 most frequent words (mostly stop words) from the column features.

We then replace every entry in the sparse frequency matrix by its pointwise mutual information (PMI) (Church and Hanks, 1990; Turney, 2001) resulting in $\boldsymbol{X}$. We factorize the matrix $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ using singular value decomposition (SVD) (Golub and Van Loan, 1996). Finally, we obtain a reduced dimensional representation of words from size $O(n)$ to k by selecting the first k columns of $\boldsymbol{U}$.

3.2 Skip-gram Vectors (SG)

`Word2Vec` word vector tool (Mikolov et al., 2013a) is currently the fastest and the most used tool for obtaining word vector representations from an unlabeled corpus.[2] Every word in the skip-gram model is represented by its Huffman code (Huffman, 1952). In this model, each current word is used as an input to a log-linear classifier with continuous projection layer and words within a certain range before and after the word are predicted.

3.3 Global Context Vectors (GC)

These word vectors (Huang et al., 2012) have been created using a neural network model which not only takes the local context of the word into account but also uses global features extracted at the document level to further enrich the vector quality.[3]

3.4 Multilingual Vectors (Multi)

These word vectors (Faruqui and Dyer, 2014) have been created using spectral learning where the authors first create monolingual word vectors by performing SVD on a monolingual word co-occurrence matrix and then use canonical correlation analysis (CCA) on pairs of vectors from different languages to obtain improved vector representations.[4]

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

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

[4] `http://www.wordvectors.org/web-eacl14-vectors/de-projected-en-512.txt.gz`

3.5 Log-bilinear Vectors (LBL)

The log bilinear language model (Mnih and Teh, 2012) predicts a word given its context where every word If h is set of context words, then the association between the target word and the context is defined as $s(w, h; Q) = \sum_{w_i \in h} q_w^\top q_{w_i} + b_{w_i}$. Thus, the probability of observing w given h is:

$$p(w|h; Q) = \frac{\exp(s(w, h; Q))}{\sum_{i=1}^{n} \exp(s(w_i, h; Q))} \quad (4)$$

Since, it is very costly to marginalize the socre 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.

4 Semantic Lexicons

We use three different semantic lexicons to evaluate their utility in improving the word vectors.

4.1 Paraphrase Database

The paraphrase database (PPDB) (Ganitkevitch et al., 2013) is a semantic lexicon containing more than 220 million paraphrase pairs of English, including 8 million lexical paraphrases, i.e. paraphrases of length 1. It has been constructed by pivoting words across multiple languages. The key concept behind obtaining lexical paraphrases through pivoting is that words of one language that are aligned to the same word in a different language should be synonymous. For example, if the words *jailed* and *imprisoned* map to the same word in another language, it may be reasonable to assume they have the same meaning. For our experiments, for each lexical paraphrase in PPDB, there exists an edge in our graph. 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 size for our experiments that produces a graph of $103,000$ nodes and $230,000$ edges[5]. Since, PPDB is an automatically created ontology, it has confidence score for a word being paraphrased into another word. We conduct two sets of experiments using the PPDB: (1) All the edge weights are equal, (2) Different edge weights for each neighbor (cf. §6).

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

4.2 WordNet

WordNet (Miller, 1995) is a large, hand-annotated 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 these synonym sets. 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* has a synonym *canine*, a hypernym *puppy* and a hyponym *animal*. We perform two different experiments with WordNet where a given word is connected to its: (1) synonyms, (2) synonyms, hypernyms and hyponyms. We get a graph of $148,000$ nodes and $560,000$ edges.

4.3 FrameNet

FrameNet (Fillmore et al., 2003) is a rich linguistic resource containing information about lexical and predicate-argument semantics in English. Grounded in the theory of frame semantics, it suggests a semantic representation that blends word-sense disambiguation and semantic role labeling. The FrameNet lexicon (Baker et al., 1998) is a taxonomy of manually identified general-purpose frames for English. Listed in the lexicon with each frame are several lemmas that can denote the frame or some aspect of it. We use this grouping of words in a particular frame as evidence that these words are semantically related and connect all words in one frame to each other in our word graph. For example, the frame `CAUSE CHANGE POSITION ON A SCALE` contains words *push, raise* and *growth*. We get a graph of $11,000$ nodes and $210,000$ edges.

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.

5.1 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) that contains 3000 word pairs which have been sampled from words that occur at least 700 times in a large web corpus.

We calculate similarity between a given pair of words by the *cosine* similarity between their corresponding vector representation. We then report Spearman's rank correlation coefficient (Myers and Well, 1995) between the rankings produced by our model against the human rankings.

5.2 Syntactic Relations (SYN-REL)

Mikolov et al. (2013b) present a new syntactic relation dataset composed of analogous word pairs. It contains pairs of tuples of word relations that follow a common syntatic relation. For example, in *walking* and *walked*, the second word is the past tense of the first word. There are nine such different kinds of relations: adjective-adverb, opposites, comaparative, superlative, present-participle, nation-nationality, past tense, plural nouns and plural verbs. Overall there are 10675 such syntactic pairs of word tuples.

The task here is to find a word *d* that best fits the following relationship: *a : b :: c : d* given *a, b* and *c*. We use the vector offset method described in Mikolov et al. (2013a) that computes the vector $q = q_a - q_b + q_c$ where, and return the vector q_w from the whole vocabulary which has the highest cosine similarity to q.

5.3 Synonym Selection (TOEFL)

The synonym selection task is to select the semantically closest word to a target from a list of candidates. The dataset we use on this task is the TOEFL dataset (Landauer and Dumais, 1997) which consists of a list of target words that appear with 4 candidate lexical substitutes each. 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.

5.4 Sentiment Analysis (SA)

Socher et al. (2013) have created a treebank which contains sentences annotated with fine-grained sentiment labels on both the phrase and sentence level and show that compositional vector space models can be used to predict sentiment at these levels with high accuracy. These sentences are movie review excerpts originally collected by Pang and Lee (2005). The coarse-grained treebank, containing only positive and negative classes has been split into training, development and test datasets containing 6920, 872 and 1821 sentences respectively. We train a logistic regression classifier with $L2$ regularization on the average of the word vectors of a given sentence to predict the coarse-grained sentiment tag at the sentence level.

6 Experiments & Results

We train word vectors of length 80 for the LSA (§3.1), log bilinear (§3.5) and the skip-gram (§3.2) vectors. The corpus used was the monolingual English news corpus from WMT-2011.[6] After normalization the corpus contained 360 million word tokens and $180,000$ word types. For the global context (§3.3) and the Multilingual (§3.4) vectors we use the word vectors available on the website. We ran experiments showing that our model can be used to retrofit the word vectors both as a post-processing step or during training.

6.1 Retrofitting by Post-processing

We use equ. 2 to optimize word vectors using different semantic lexicons under different settings as described below.

PPDB. There exists an edge between a word and all its paraphrases. For every word w_i, we set $\alpha_i = 1$ and the edge weight for every neighbor $\forall j, \beta_{ij} = 1/N$, where N is the number of neighbors of the word. This makes sure that the optimized word vector is still not very far from its originally observed estimate.

wtPPDB. PPDB has weights associated with every paraphrase word pair $p(w_i|w_j)$ and $p(w_j|w_i)$ which are the conditional probabilities of observing the paraphrase given the word and vice versa obtained empirically during the construction of the PPDB. We define $\forall j, \beta_{ij} = p(w_i|w_j)/2 + p(w_j|w_i)/2$ and keep $\alpha_i = 1$ as in the previous experiment.

[6] http://www.statmt.org/wmt11/

WN. There exists an edge between a word and all of its synonyms. For every word w_i, we set $\alpha_i = 1$ and the edge weight for every neighbor $\forall j, \beta_{ij} = 1/N$, where N is the number of neighbors of the word.

WN++. There exists an edge between a word and all of its synonyms, hypernyms and hyponyms. For every word w_i, we set $\alpha_i = 1$ and the edge weight for every neighbor $\forall j, \beta_{ij} = 1/N$, where N is the number of neighbors of the word.

FN. There exists an edge between all the words that evoke the same semantic frame. For every word w_i, we set $\alpha_i = 1$ and the edge weight for every neighbor $\forall j, \beta_{ij} = 1/N$, where N is the number of neighbors of the word.

Results. Figure 2 shows the absolute improvements in both the spearman's correlation ratio and the accuracy obtained on different tasks over the baseline. Every row of plots shows one semantic lexicon being used for retrofitting. The different vector models are shown in different colors.

We see that all the lexicons are helpful in improving the Spearman's correlation ratio in the word similarity tasks. However, FrameNet is either not improving the results or even making them worse in many cases, for example the skip-gram and multilingual vectors. In the syntactic relation task (§5.2), we get huge improvements of the order of 10 absolute points in accuracy for all ontologies except for FrameNet. For the extrinsic sentiment analysis task (§5.4) we get improvements using all the ontologies and achieve the highest improvement of absolute 1.4 points in accuracy for the multilingual vectors over the baseline (§3.4). is statistically significant ($p < 0.01$) according to a McNemar's test (Dietterich, 1998).

Interestingly, **wtPPDB** is almost as good as or even worse than the uniformly weighted **PPDB** setting which signifies that PPDB already has a highly confident set of paraphrases which are of equal quality. Also, the reason that FrameNet does not perform as well as the other ontologies may be because words of significantly different meaning can evoke the same frame. For example, *push* and *growth* are two words of different meanings but they evoke the same frame as described in §4.3. Overall, we observe that **PPDB** helps in improving the quality of the word vectors irrespective of the type of task and the word vector model.

Lexicon Ensemble. We now examine if we can combine multiple ontologies together in a way which can lead to performance gain more than either of these ontologies alone. We perform two experiments, in the first experiment we take a union of the sets of edges of the two lexicons and in the second experiment we take the intersection of the sets of edges. We then perform retrofitting of vectors using the new lexicon and carry out the evaluation. As can be expected, taking union of ontologies performed better than taking intersection and hence we only show the union results in the bottom row of figure 2. We only explore the union of WordNet and PPDB and exclude FrameNet as it did not give positive results on its own.

6.2 Retrofitting while Training

We now discuss the experiments of retrofitting while training the word vectors. Note that for this we need to alter the training algorithm and hence we conducted experiments only with the log bilinear model (§3.5) for which we implemented the model training ourselves. We conduct experiments for both **XYZ** and **ABC** algorithms of MAP estimation. For the **XYZ** method we update the prior every $k = 100,000$ words[7] and test for different values of $\gamma \in [1, 0.1, 0.01]$. For the **ABC** method, we update the word vectors using equ. 2 every $k \in [25, 50, 100]$ million words.

Table 1 shows the results of improvement while training word vectors of length 80. We show experimental results with only PPDB (§4.1) as the ontology. For **XYZ** $\gamma = 0.01$ performs the best although other values of γ also yield very close results. For **ABC** $k = 50$ million perform the best although all other values of k also perform better than the baseline. Overall we see that the both the methods lead to improvement in performance across all tasks however **ABC** is relatively better in performance.

6.3 Multilingual Evaluation

We tested our method on three different languages other than English: German, French and Spanish. We used the Universal WordNet (de Melo and Weikum, 2009), which is an automatically constructed multilingual lexical knowledge base based on WordNet.[8] It contains words connected

[7] Experiments with $k \in [10000, 50000]$ yielded almost similar results.

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

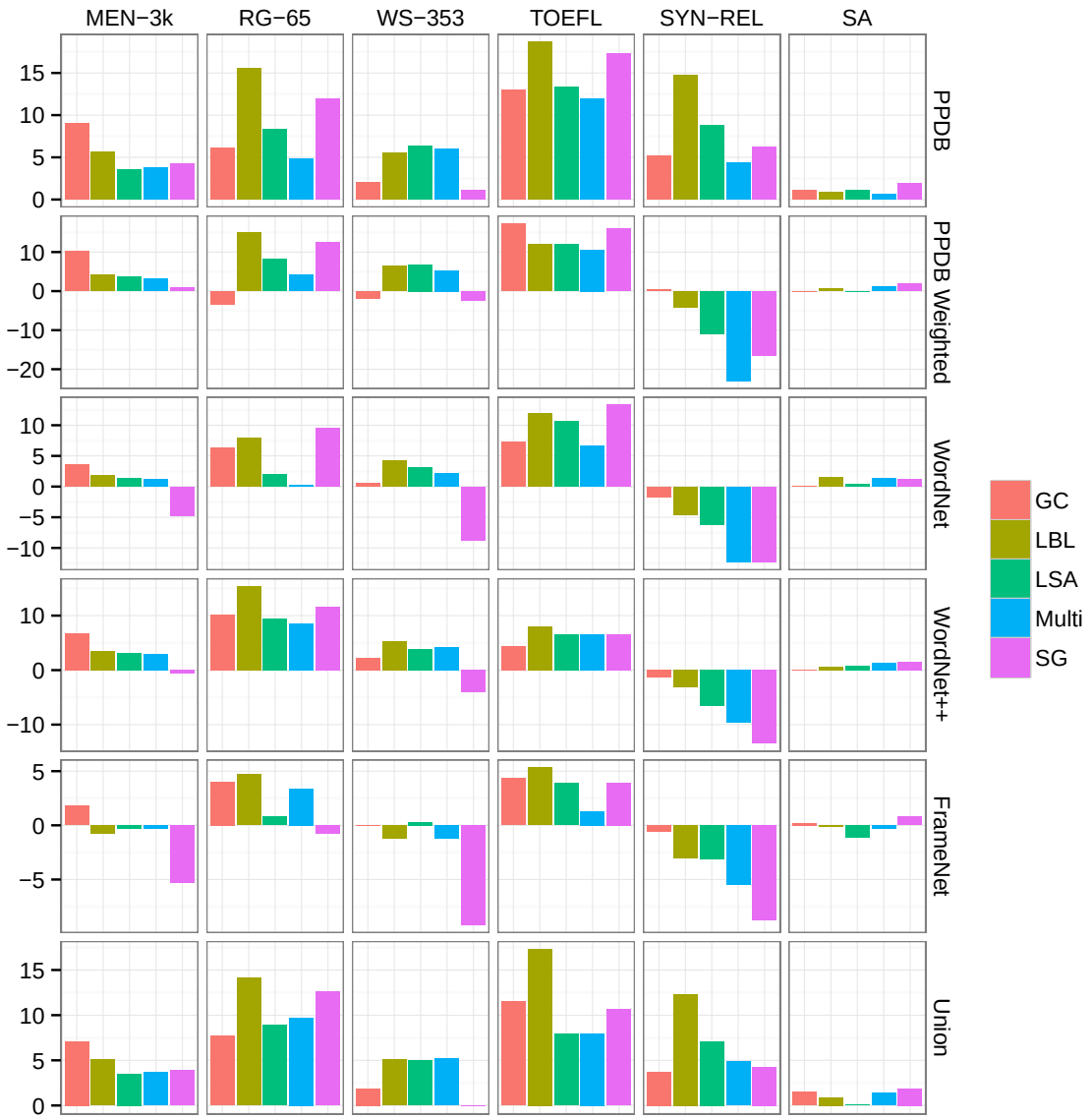

Figure 2: Absolute improvement in the Spearman's correlation ratio (3 left columns) and accuracy (3 right columns) using different semantic lexicons.

Method	k/γ	WS-353	RG-65	MEN	SYN-REL	TOEFL	SA
Baseline	$k = \infty, \gamma = 0$	53.6	42.7	58.0	31.5	66.7	72.5
XYZ	$\gamma = 1$	54.2	46.9	57.6	32.1	66.6	**73.7**
	$\gamma = 0.1$	54.0	50.8	58.7	32.2	65.3	73.3
	$\gamma = 0.01$	**55.3**	**52.2**	**58.7**	**33.4**	**69.3**	72.9
ABC	$k = 100m$	57.2	61.1	**61.8**	**36.3**	78.7	73.8
	$k = 50m$	**58.0**	**62.2**	61.4	32.1	85.3	**74.4**
	$k = 25m$	56.3	60.8	58.5	27.8	**88.0**	73.3

Table 1: Spearman's correlation (3 left columns) and accuracy (3 right columns) on different tasks. Bold indicates best result across all vector types.

via different lexical relations to other words both in and across languages. For a word in a language we only consider links going to the other words in the same language to construct the word graph. We first train word vectors for these three languages and then improve them using the ontological information.

For German, French and Spanish we constructed word graphs containing around 87000, 49000 and 31000 word nodes being connected

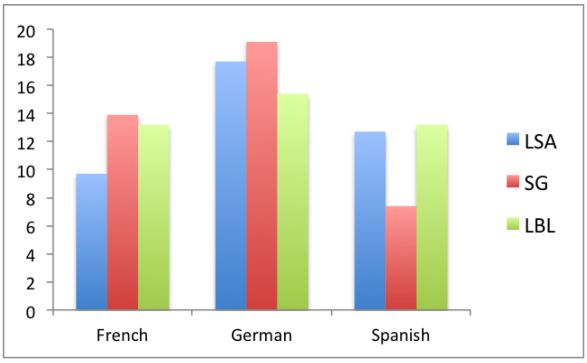

Figure 3: Improvement in spearman's correlation for word similarity evaluation using the retrofitted vectors from different word vector models on French (WS-353), German (RG-65) and Spanish (MC-30).

to other nodes via 200000, 100000 and 52000 edges respectively. Since not many word similarity evaluation benchmarks are available for other languages we tested our baseline and improved vectors on one benchmark per language. We used RG-65 (Gurevych, 2005), WS-353 (Joubarne and Inkpen, 2011) and MC-30 (Hassan and Mihalcea, 2009) for German, French and Spanish respectively. We trained three different types of vectors: LSA (§3.1), Skip-Gram (§3.2) and LBL (§3.5) for all the three languages of length 80 while keeping all other parameters same as previously described. We again used the WMT-2011 monolingual news corpus for all the languages and evaluate word similarity on these tasks before and after enrichment. Figure 3 shows the results that strongly indicate that our method generalizes across languages. We get high improvements in the Spearman's correlation coefficient on the word similarity tasks for the three languages for three different kinds of vectors.

7 Sense Specific Vectors?

Most word vector models assign one vector to each word type ignoring the possibility that the same word type may have more than one meaning. Should an obviously polysemous word like *bank* really be represented using the same structure (i.e, a single vector) as a more monosemous word *anger*? Two previous efforts have attempted to learn sense-specific vectors by clustering contexts to discriminate word senses into multiple prototype vectors (Reisinger and Mooney, 2010; Huang et al., 2012). However, the resulting vectors lack interpretability and cannot be matched to senses present in a lexicon, for ex., WordNet.

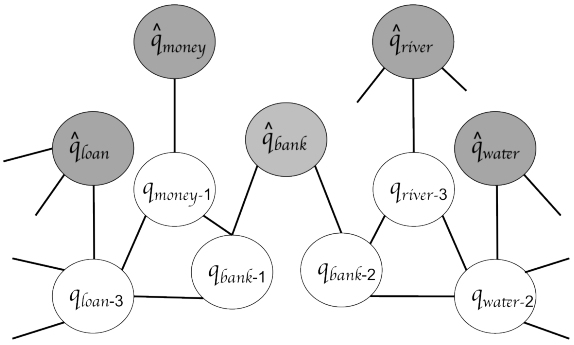

Figure 4: Word graph for inducing sense-specific vectors from a sense-agnostic word vector model.

We augment our retrofitting framework (§2) using WordNet that provides information about the number of senses for a particular word and its synset (synonym set). We introduce an unobserved node for every sense of a word and connect all of them to one observed node in the graph, which represents the word vector obtained for that word from a word vector model. We then connect each of these nodes to other words in its synset in the regular manner. For example, in figure 4, for the word *bank*, if we have two senses in the WordNet, we create two unobserved nodes each connected to the observed node representing the word vector $\hat{q}_{bank}$ and also connected to other words in their respective synset. Now performing belief propagation on this graph as a simple post-processing step (§2.1) gives us the sense specific vector corresponding to each sense in the WordNet. Intuitively the sense-laebeled vectors "pull apart" the components of the sense-agnostic vectors.

In table 2 we present some analysis that qualitatively attempts to evaluate the nature of sense-specific word vectors. The table shows the three most similar words of an ambiguous word in standard SG model (§3.2) in comparison with the three most similar words of different word senses of the same word in the disambiguated SG model. The standard vector models are either dominated by the most frequent sense or they mix multiple senses of a word. On the other hand, the disambiguated vectors appear to capture sense specificity of less frequent senses successfully. The examples in table 2, while chosen for their prominence in evidencing the phenomenon, are fairly representative of the whole vector space in gen-

Word or Word Sense	Top 3 Most Similar Words or Word Senses		
hanging hanging%1:04:01:: (act of suspending something) hanging%1:06:00:: (decoration hung on a wall)	hung shoring%1:04:00:: tapestry%1:06:00::	dangled support%1:04:01:: braid%2:35:01::	hangs suspension%1:04:00:: smock%2:36:00::
crossbar crossbar%1:06:01:: (game equipment horizontal bar) crossbar%1:06:02:: (horizontal bar across something)	left-foot header%1:06:00:: shackle%1:06:01::	left-footed equalizer%1:06:01:: heaver%1:06:00::	header goalie%1:18:00:: bar%1:06:00::
climber climber%1:18:00:: (someone who climbs as a sport) climber%1:20:00:: (a vine or climbing plant)	climbers lifter%1:18:00:: woodbine%1:20:02::	skier swinger%1:18:01:: brier%1:20:02::	Loretan sharpshooter%1:18:01:: kiwi%1:20:00::

Table 2: The top 3 most similar words or word senses for some polysemous words. WordNet definitions for word senses are given in brackets where appropriate.

eral. We propose to carry out extensive quantitative evaluation of the sense-specific vectors as future work.

8 Related Work

In terms of methodology, the approach we propose is conceptually similar to previous work that leverages 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 induction (Das and Smith, 2011) both of which use belief propagation in a manner similar to ours to obtain labels for unknown words.

The use of lexical semantic information in training word vectors has been limited. Recently, Yu and Dredze (2014) used semantic knowledge to improve the Word2Vec (Mikolov et al., 2013a) embeddings in a joint training model similar to our retrofitting while training approach (§2.2). Alfonseca and Manandhar (2002) attempt to extend an ontology such as WordNet with unsupervised and domain-specific information that is yielded by distributional vectors. Agirre et al. (2001) enrich the WordNet ontology with topic rather than raw distributional signatures and Agirre et al. (2013) use the WordNet graph to perform random-walks that guide a word-sense disambiguation process.

Word vectors have also shown to improve using cross lingual information both during training (Zou et al., 2013; Hermann and Blunsom, 2014) and as a post-processing operation (Faruqui and Dyer, 2014). There have also been disparate attempts at enhancing distributional semantic representations with various sources of non-linguistic information such as images (Bruni et al., 2011) and experiential data (Andrews et al., 2009).

Our approach of learning better word vector representations can be seen as an instance of graph based semi-supervised learning (Zhu, 2005; Talukdar and Pereira, 2010) where the supervision is derived from the explicit word relations in Semantic Lexicons. Graph based learning has also been employed in machine translation (Alexandrescu and Kirchhoff, 2009; Saluja et al., 2014), 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 to improve word vectors (obtained from different models of word representations) using either automatically or human constructed semantic lexicons that have explicit information about word relations. We have shown that retrofitting the word vectors to semantic lexicons is useful and can be performed both during training the word vectors or as a post-processing operation. We validated the applicability of our method across a number of languages and showed that performance improvement can be obtained on different types of evaluation tasks.

References

Eneko Agirre, Olatz Ansa, Eduard Hovy, and David Martinez. 2001. Enriching wordnet concepts with topic signatures. *arXiv preprint cs/0109031*.

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*, NAACL '09, pages 19–27, Stroudsburg, PA, USA.

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2013. Random walks for knowledge-based word sense disambiguation.

Andrei Alexandrescu and Katrin Kirchhoff. 2009. Graph-based learning for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 119–127, Stroudsburg, PA, USA. Association for Computational Linguistics.

Enrique Alfonseca and Suresh Manandhar. 2002. Extending a lexical ontology by a combination of distributional semantics signatures. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, pages 1–7. Springer.

Mark Andrews, Gabriella Vigliocco, and David Vinson. 2009. Integrating experiential and distributional data to learn semantic representations. *Psychological review*, 116(3):463.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 2006. Label propagation and quadratic criterion. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, pages 193–216. MIT Press.

Elia Bruni, Giang Binh Tran, and Marco Baroni. 2011. Distributional semantics from text and images. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 22–32. Association for Computational Linguistics.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics.

Bob Carpenter. 2008. Lazy sparse stochastic gradient descent for regularized multinomial logistic regression. technical report, alias-i. available at http://lingpipe-blog.com/ lingpipe-white-papers.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Mark Culp and George Michailidis. 2008. Graph-based semisupervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):174–179.

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 the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pages 513–522, New York, NY, USA. ACM.

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*.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.

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

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden, April. Association for Computational Linguistics.

Charles Fillmore, Christopher Johnson, and Miriam Petruck. 2003. Background to framenet. *International Journal of Lexicography*, 16(3):235–250.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 406–414, New York, NY, USA. ACM Press.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia, June. Association for Computational Linguistics.

Andrew B. Goldberg and Xiaojin Zhu. 2006. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, pages 45–52, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gene H. Golub and Charles F. Van Loan. 1996. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

Iryna Gurevych. 2005. Using the structure of a conceptual network in computing semantic relatedness. In *Proceedings of the Second International Joint Conference on Natural Language Processing*, IJCNLP'05, pages 767–778, Berlin, Heidelberg. Springer-Verlag.

Samer Hassan and Rada Mihalcea. 2009. Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1192–1201, Stroudsburg, PA, USA. Association for Computational Linguistics.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. *arXiv preprint arXiv:1404.4641*.

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 the 50th ACL: Long Papers-Volume 1*, pages 873–882.

David Huffman. 1952. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE*, 40(9):1098–1101, September.

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 the 24th Canadian Conference on Advances in Artificial Intelligence*, Canadian AI'11, pages 216–221, Berlin, Heidelberg. Springer-Verlag.

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

Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Comput. Linguist.*, 29(1):53–72, March.

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*, pages 211–240.

Joel Lang and Mirella Lapata. 2011. Unsupervised semantic role induction with graph partitioning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1320–1331, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 the 2013 Conference of the NAACL: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1751–1758.

Jerome L. Myers and Arnold D. Well. 1995. *Research Design & Statistical Analysis*. Routledge, 1 edition, June.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Comput. Linguist.*, 33(2):161–199, June.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2010)*, pages 109–117.

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

Avneesh Saluja, Hany Hassan, Kristina Toutanova, and Chris Quirk. 2014. Graph-based semi-supervised learning of translation models from monolingual data. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

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 the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA, October. Association for Computational Linguistics.

Amarnag Subramanya, Slav Petrov, and Fernando Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 167–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Partha Pratim Talukdar and Fernando Pereira. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1473–1481, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th ACL*, ACL '10, pages 384–394, Stroudsburg, PA, USA.

Peter D. Turney. 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, London, UK, UK. Springer-Verlag.

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

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Association for Computational Linguistics (ACL)*.

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

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