Improving Vector Space Word Representations Using Prior Semantic Beliefs

Abstract

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

1 Introduction

The distributional hypothesis of Firth (1957) *"you shall know a word by the company it keeps"* has led to a large interest in deriving various semantic representations of words in terms of their frequent contextual patterns. One such representations is the vector space model of word meaning representation that encodes a word in a vector of real numbers. 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).

Data-driven learning of vector-space word embeddings that capture lexico-semantic properties is a technique of central importance in natural language processing. 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) to arrive at vector representations that capture co-occurence tendencies and meanings, it is possible to construct high-quality semantic vectors. Variants of not only using co-occurrence statistics from monolingual corpora but looking at alignments of words across languages (Zou et al., 2013; Hermann and Blunsom, 2014; Faruqui and Dyer, 2014) have been shown to perform better than their monolingual counterparts. Another variant of constructing the word vectors considers the co-occurrence of words as determined by their dependency syntax (Padó and Lapata, 2007).

In a similar spirit, we propose that word vector representations can be further improved by looking at the neighboring words in a lexical ontology. Lexical ontologies 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. These relations in turn determine the semantic association between the words. This is an obvious of source of information that has not been exploited in creating word vectors so far.

We present a graph-based learning framework which uses the lexical ontologies to create a word graph (§2) where semantically associated words are connected to each other. We then perform belief propagation on the graph which enforces similar words to have similar word vector representations. Our method of incorporating prior semantic beliefs can be used both while training (§6.3) the word vectors or as a post-processing step (§6.1). We show that our method works well with a large number of different types of word vector models (§3) and gives substantial improvements on a variety of vector evaluations tasks (§5) while using different kinds of lexical ontologies (§4) across dif-

ferent languages (§6.4).

2 Graph Encoding of Prior Beliefs

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 some relation. For now we consider all the word types to be unambiguous in meaning, although later we will see how our model also lets us derive word embeddings for different senses of word types. For the sake of simplicity consider $\forall i, w_i$ to be unambiguous word-types, although the model we propose is flexible enough to accommodate polysemy (§).

Then, given a set of vectors $\hat{Q} = \{\hat{q}_i | \ \forall \ w_i \in W\}$ that have been learned from any of the currently available methods to obtain word vector representations, 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 Ω by a notion of distance metric between vectors. Figure 1 shows a small word graph with such edge connections. The layer containing vectors Q and vectors $\hat{Q}$ can be seen as the unobserved and observed layers 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 improved) word representations $\hat{Q}$.

We define the distances between any two nodes as the euclidean distance between them, thus forcing similar 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 clique potential becomes:

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

where, β_{ij} is the weight of the edges connecting i, j and α is a hyper-parameter to control the degree of deviation of the word vector from its observed value. Equation 1 is a convex optimization problem and a close formed solution does exist but practically is intractable to obtain. Thus, we take the gradient of equation 1 with respect to the parameters to be optimized, namely: $\forall i, q_i$ and equating it to zero get the following update for q_i:

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

We run this procedure for 10 iterations to achieve convergence.

3 Word Vector Representations

We show that our method to improve word vectors can be used both as a post-processing step or during training. For the post-processing step, we take word vectors available from four different models and enrich them using equation 2. To show that our model can also be used while training the word vectors, we train the log-bilinear word embeddings model (§3.5).

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 $\boldsymbol{F}$ 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 $\boldsymbol{F}$ 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 $|\boldsymbol{V}|$ to k by selecting the first k columns of $\boldsymbol{U}$.

3.2 Skip-gram Vectors (SG)

`Word2Vec` word vector tool (Mikolov et al., 2013) is currently the fastest and the most used tool for obtaining word vector representations from an unlabeled corpus[1]. Every word in the Skip-gram model is represented by its Huffman code (Huffman, 1952). In this model, each current

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

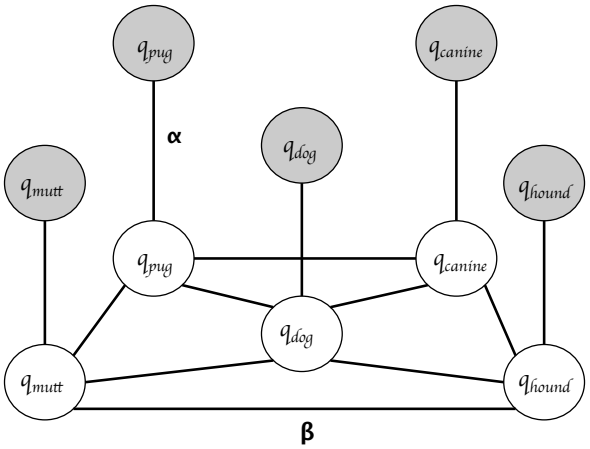

Figure 1: Word graph with edges between related words showing the observed and the inferred word vector representations.

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[2].

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 cooccurrence matrix and then use canonical correlation analysis (CCA) on pairs of vectors from different languages to obtain improved vector representations.

3.5 Log-bilinear Vectors (LBL)

The log bilinear language model (Mnih and Teh, 2012) predicts a word given its context where every word w is represented by a word vector q_w and a bias b_w which constitute the parameters θ of the model. If h is set of context words, then the association between the target word and the context can be written as $s_\theta(w,h) = \sum_{i=1}^{h} q_w^\top q_{w_i} + b_{w_i}$. Thus, the probability of observing w given h is:

$$P_\theta(w|h) = \frac{\exp(s_\theta(w,h))}{\sum_{i=1}^{n}\exp(s_\theta(w_i,h))} \quad (3)$$

where $|V|$ is the size of the vocabulary. Since, it is very costly to marginalize the score over the whole vocabulary we use[3] *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.

Improving during Training. Training LBL vectors requires performing MLE using equation 3. We can instead perform MAP estimation if we encode semantic beliefs as prior of the parameters θ. We define the prior as:

$$P(\theta) = \exp(-\gamma \sum_{i=1}^{n} \sum_{ij \in E_\Omega} \|q_{w_i} - q_{w_j}\|^2) \quad (4)$$

We take gradient of equ. 4 with respect to θ and add it to the NCE gradient to perform MAP. However, since computing gradient of equ. 4 is linear in the vocabulary size n, we use lazy updates (Carpenter, 2008) every k words during training. We call this the **MAP** method of improvement during training.

Note that the NCE objective is non-convex while the graph based objective (equ. 2) is convex. We alternate between these two objectives while training the word vectors by using equation 2 to re-estimate the word vectors after we visit every k

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

[3] We implemented this model ourselves as the code is not publicly available

words during training. This method of alternating between a convex and a non-convex objective[4] is a kind of simulated annealing that helps in overcoming local optima (**[citation reqd. –MFAR]**). We call this method the **SWAP** method of improvement during training.

4 Lexical Ontologies

We use three different lexical ontologies to separately evaluate their utility in improving the word vectors and then merge them all together to act as a single large ontological resource.

4.1 Paraphrase Database

The paraphrase database (PPDB) (Ganitkevitch et al., 2013) is an ontology of more than 220 million paraphrase pairs, 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 *thrown into jail* 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 contains approximately $103,000$ paraphrases[5].

Since, the PPDB is a statistically created ontology, it has weights associated with the confidence of a source word being paraphrased into a target 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).

4.2 WordNet

WordNet (Miller, 1995) is a large, hand-annotated lexical ontology 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. Since, WordNet is a carefully hand crafter resource, we at the moment weigh all our edges to be of weight 1. We get a mapping of approximately $143,000$ words being connected to other words from WordNet.

4.3 FrameNet

FrameNet (Fillmore et al., 2003) is a rich linguistic resource containing considerable 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 a hint 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 mapping of $10,822$ words being connected to others from FrameNet.

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

[4] An improvement in log-likelihood of the language model was also observed over the pure non-convex optimization case.

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

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. (2013) 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. (2013) that computes the vector $\boldsymbol{y} = \boldsymbol{x}_a - \boldsymbol{x}_b + \boldsymbol{x}_c$ where, $\boldsymbol{x}_a, \boldsymbol{x}_b$ and $\boldsymbol{x}_c$ are word vectors of a, b and c respectively and returns the vector $\boldsymbol{x}_w$ from the whole vocabulary which has the highest cosine similarity to $\boldsymbol{y}$:

$$\boldsymbol{x}_w = \arg\max_{\boldsymbol{x}_w} \frac{\boldsymbol{x}_w \cdot \boldsymbol{y}}{|\boldsymbol{x}_w| \cdot |\boldsymbol{y}|} \tag{5}$$

It is worth noting that this is a non-trivial $|V|$-way classification task where V is the size of the vocabulary.

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. These sentences are movie review excerpts originally collected by Pang and Lee (2005). They show that compositional vector space models can be used to predict sentiment at these levels with high accuracy. 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 SG (§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 do not train our own models and simply use the word vectors available on the website. We ran experiments showing that our model can be used to improve the word vectors both during training or as a post-processing step.

6.1 Improving by Post-processing

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

PPDB. 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 original 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. We keep all the settings as in **PPDB** except for using synonyms from WordNet as ontological edges.

WN++. We keep all the settings as in **WN** except for using all the synonyms, hypernyms and hyponyms of a given word as ontological edges.

FN. We keep all the settings as in **PPDB** except for using FrameNet as the ontological resource. There is an edge between all the words evoking a particular frame.

Results. Table 1 shows the results. We see that all the word ontologies 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).

Interestingly, **wtPPDB** is almost as good as the uniformly weighted **PPDB** setting which signifies that PPDB already has a highly confident set of paraphrases which are of equal quality. Maybe as we increase the size of PPDB from XL to XXXL these weights would become more significant as the larger version of PPDB has smaller confidence estimates (Ganitkevitch et al., 2013). Also, the reason that FrameNet does not perform as good 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 entirely different meaning but they evoke the same frame as describe in §4.3.

6.2 Ontology Ensemble

After evaluating the performance of word vectors enriched using individual ontologies we observe that PPDB and WordNet provide good evidence of lexical similarity and relatedness. 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. Let Ω_{new} be the combine ontology and Ω_1 and Ω_2 be the individual ones. For every edge e_{ij} between words w_i and w_ij, we let $e_{ij} \in \Omega_{new}$ if e_{ij} belongs to both Ω_1 and Ω_2 or either of them, in effect taking a union and intersection of the ontologies. We then perform enrichment of vectors using the new ontology Ω_{new} 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 Table 1. We only explore the union of WordNet and PPDB and exclude FrameNet as it did not give positive results on its own.

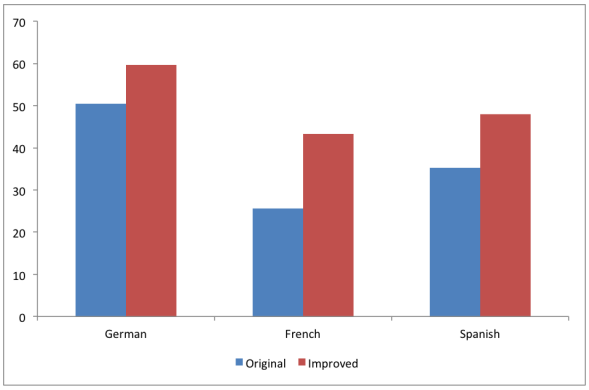

Figure 2: Spearman's correlation for word similarity evaluation on different languages.

6.3 Improving while Training

We discuss experiments conducted using both **MAP** and **SWAP** methods of improving word vectors using prior beliefs while training. For the **MAP** 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 **SWAP** method, we update the word vectors using equ. 2 every $k \in [25, 50, 100]$ million words.

Table 2 shows the results of improvement while training for vectors of length 80. We show experimental results with only PPDB (§4.1) as the ontology for brevity. For **MAP** $\gamma = 0.01$ performs the best although the other values of γ also yield very close results. For **SWAP** $k = 50$ million perform the best although all other values of k also perform better than the baseline. Overall we see that the simpler method **SWAP** beats **MAP** hands down on all the tasks with a big margin.

6.4 Multilingual Evaluation

We tested our method on three different languages other than English: German, French and Spanish. We used the Universal WordNet (de Melo

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

Vector	Ontology	Dim	WS-353	RG-65	MEN	SYN-REL	TOEFL	SA
LSA	–	80	61.4	66.9	70.9	34.6	66.7	77.9
	PPDB	80	67.7	75.2	**74.5**	**43.4**	**80.0**	**79.0**
	PPDBwt	80	**68.2**	75.2	74.5		78.6	77.9
	WN	80	64.5	68.9	72.3	28.4	77.3	78.3
	WN++	80	65.3	76.4	74.1	28.0	73.3	78.8
	FN	80	61.7	67.7	70.6	31.5	70.6	76.8
	Union	80	67.4	**76.5**	73.9		73.3	77.6
SG	–	80	63.9	54.6	64.6	45.5	66.7	74.5
	PPDB	80	**65.0**	66.5	**68.8**	**51.8**	**84.0**	76.4
	PPDBwt	80	61.6	67.1	65.5		82.6	76.4
	WN	80	55.1	64.1	59.8	33.2	80.0	75.7
	WN++	80	59.8	66.3	63.9	32.1	73.3	76.1
	FN	80	54.7	53.8	59.3	36.8	70.6	75.3
	Union	80	63.8	**67.2**	68.5		77.3	**76.4**
GC	–	50	62.3	62.8	31.3	10.9	60.8	67.8
	PPDB	50	64.3	68.9	40.3	**16.1**	73.9	**68.9**
	PPDBwt	50	60.4	59.5	**41.6**		**78.2**	67.7
	WN	50	62.9	69.2	34.9	9.2	68.1	67.8
	WN++	50	**64.6**	73.0	38.0	9.6	65.2	67.9
	FN	50	62.3	66.8	33.1	10.3	65.2	68.0
	Union	50	64.2	**70.5**	38.4		72.4	68.7
Multi	–	512	68.1	75.5	75.8	45.5	84.0	81.0
	PPDB	512	**74.1**	79.5	79.1	**49.8**	**96.0**	81.5
	PPDBwt	512	73.4	80.3	**79.6**		94.6	82.1
	WN	512	70.3	75.7	77.0	33.2	90.6	82.4
	WN++	512	72.4	84.0	78.7	36.0	90.6	82.4
	FN	512	66.9	78.9	75.5	40.1	85.3	80.7
	Union	512	73.3	**85.2**	**79.5**		92.0	**82.4**
LBL	–	80	53.6	42.7	58.0	31.5	66.7	72.5
	PPDB	80	59.1	**58.3**	**63.7**	**46.2**	**85.3**	73.4
	PPDBwt	80	**60.1**	57.8	62.1		78.6	73.2
	WN	80	57.9	50.6	59.9	26.8	78.7	74.0
	WN++	80	58.9	58.1	61.5	28.3	74.7	73.1
	FN	80	52.4	47.4	57.2	28.4	72.0	72.4
	Union	80	58.7	56.8	63.1		84.0	**73.4**

Table 1: Enrichment as Post-processing of various word embeddings using different lexical ontologies. Spearman's correlation (left) and accuracy (right) on different tasks. Bold indicates best result across all vector types.

Method	k/γ	WS-353	RG-65	MEN	SYN-REL	TOEFL	SA
Baseline	$k = \infty, \gamma = 0$	53.6	42.7	58.0	31.8	66.7	72.5
MAP	$\gamma = 1$	54.2	46.9	57.6		66.6	**73.7**
	$\gamma = 0.1$	54.0	50.8	58.7		65.3	73.3
	$\gamma = 0.01$	**55.3**	**52.2**	**58.7**		**69.3**	72.9
SWAP	$k = 100m$	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 2: Enrichment while Training of LBL word embeddings using PPDB. Spearman's correlation (left) and accuracy (right) on different tasks. Bold indicates best result across all vector types.

and Weikum, 2009), which is an automatically constructed multilingual lexical knowledge base based on WordNet as the ontological resource[8]. It contains words connected 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.

For German, French and Spanish we were able to retrieve around 87000, 49000 and 31000 words being connected to other words respectively. 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 LSA vectors (§3.1) of length 80 on WMT-2011 monolingual news corpus for all the languages and evaluate word similarity on these tasks before and after enrichment

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

as shown in figure 2. The results indicate that our method generalizes across languages.

7 Qualitative Analysis

To understand how ontological evidence leads to better results in semantic evaluation tasks, we plot the representations obtained in §3.1 both before and after enrichment of the words that showed significant change in euclidean distance between the two vectors. We project the vectors onto $\mathbb{R}^2$ using the t-SNE tool (van der Maaten and Hinton, 2008) (cf. figure 3). Interestingly, most of these words are either minor spelling variations or a rarely used similar word to one of the frequent words. For example, *amerika* and *afganistan* are mis-spelled forms of *america* and *afghanistan* recpectively. Also, *behind-the-scenes* and *attaboy* are rare frequency sysnonyms of the words *backdoor* and *bravo* respectively. We see that these synonyms words are closer after enrichment (in red) than originally (in green).

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). Most notably, Das and Smith (2011) construct a factor graph of a semantic lexicon that is used to generalize and apply semantic frames to previously unseen types. Graph based belief propagation has also been used to induce POS tags (Subramanya et al., 2010). Our approach is closest to Das and Petrov (2011) who induce POS tag distribution for words in a resource poor language through a resource rich language but differs in that we induce monolingual word vector representations instead of POS tag distributions.

The use of ontological information in training word vectors has been limited. Alfonseca and Manandhar (2002), for example, attempt to extend an ontology such as Wordnet with unsupervised and domain-specific information that is yielded by distributional vectors. In contrast, Gabrilovich and Markovitch (2009) directly construct an ontology of concepts based on the Wikipedia knowledge-graph using a technique called Explicit Semantic Analysis which heavily relies on distributional statistics. 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 lexical ontologies. Graph based learning has also been employed in 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 to improve word vectors (obtained from different models of word representations) using either automatically or human constructed lexical ontologies that have explicit information about word relations. We have shown that the ontological information is useful and it can be used 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 shown 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.

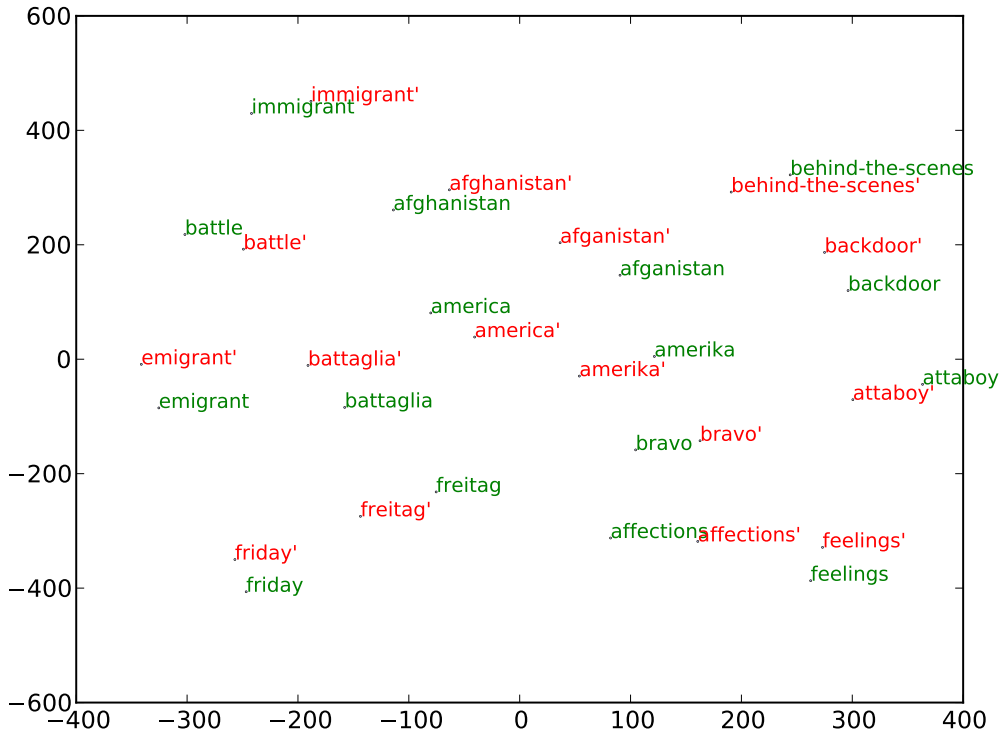

Figure 3: Synonymous words or spelling variants of words come closer after enrichment (red) than originally (green).

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.

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*.

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.

J.R. Firth. 1957. A synopsis of linguistic theory 1930-1955. *Studies in linguistic analysis*, pages 1–32.

Evgeniy Gabrilovich and Shaul Markovitch. 2009. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34(2):443.

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. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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.

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 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.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, November.

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.