Ontologically Grounded Multi-sense Representation Learning for Semantic Vector Space Models

Sujay Kumar Jauhar **Chris Dyer** **Eduard Hovy**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{sjauhar, cdyer, hovy}@cs.cmu.edu

Abstract

Words are polysemous. However, most approaches to representation learning for lexical semantics assign a single vector to every surface word type. Meanwhile, lexical ontologies such as WordNet provide a source of complementary knowledge to distributional information, including a word sense inventory. In this paper we propose two novel and general approaches for generating sense-specific word embeddings that are grounded in an ontology. The first applies graph smoothing as a post-processing step to tease the vectors of different senses apart, and is applicable to any vector space model. The second adapts predictive maximum likelihood models that learn word embeddings with latent variables representing senses grounded in an specified ontology. Empirical results on lexical semantic tasks show that our approaches effectively captures information from both the ontology and distributional statistics. Moreover, in most cases our sense-specific models outperform other models we compare against.

1 Introduction

Vector space models (VSMs) of word meaning play a central role in computational semantics. These represent meanings of words as contextual feature vectors in a high-dimensional space (Deerwester et al., 1990) or some embedding thereof (Collobert and Weston, 2008) and are learned from unannotated corpora. Word vectors in these continuous space representations can be used for meaningful semantic operations such as computing word similarity (Turney, 2006), performing analogical reasoning (Turney, 2013) and discovering lexical relationships (Mikolov et al., 2013b). They have also proved useful in downstream NLP applications such as information retrieval (Manning et al., 2008) and question answering (Tellex et al., 2003), among others.

However, VSMs remain flawed because they assign a single vector to every word, thus ignoring the possibility that words may have more than one meaning. For example, the word "bank" can either denote a financial institution or the shore of a river. The ability to model multiple meanings is an important component of any NLP system, given how common polysemy is in language. The lack of sense annotated corpora large enough to robustly train VSMs, and the absence of fast, high quality word sense disambiguation (WSD) systems makes handling polysemy difficult.

Meanwhile, lexical ontologies, such as WordNet (Miller, 1995) specifically catalog sense inventories and provide typologies that link these senses to one another. These hand-curated ontologies provide a complementary source of information to distributional statistics. Recent research tries to leverage this information to train better VSMs (Yu and Dredze, 2014; Faruqui et al., 2014), but does not tackle the problem of polysemy. Parallely, work on polysemy for VSMs revolves primarily around techniques that cluster contexts to distinguish between different word senses (Reisinger and Mooney, 2010; Huang et al., 2012), but does not integrate ontologies in any way.

In this paper we present two novel approaches to integrating ontological and distributional sources of information. Our focus is on allowing already existing, proven techniques to be adapted to produce ontologically grounded word sense embeddings. Our first technique is applicable to any sense-agnostic

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 683–693,
Denver, Colorado, May 31 – June 5, 2015.

VSM as a post-processing step that performs graph propagation on the structure of the ontology. The second is applicable to the wide range of current techniques that learn word embeddings from predictive models that maximize the likelihood of a corpus (Collobert and Weston, 2008; Mnih and Teh, 2012; Mikolov et al., 2013a). Our technique adds a latent variable representing the word sense to each token in the corpus, and uses EM to find parameters. Using a structured regularizer based on the ontological graph, we learn grounded sense-specific vectors.

There are several reasons to prefer ontologies as distant sources of supervision for learning sense-aware VSMs over previously proposed unsupervised context clustering techniques. Clustering approaches must often parametrize the number of clusters (senses), which is neither known a priori nor constant across words (Kilgarriff, 1997). Also the resulting vectors remain abstract and uninterpretable. With ontologies, interpretable sense vectors can be used in downstream applications such as WSD, or for better human error analysis. Moreover, clustering techniques operate on distributional similarity only whereas ontologies support other kinds of relationships between senses. Finally, the existence of cross-lingual ontologies would permit learning multi-lingual vectors, without compounded errors from word alignment and context clustering.

We evaluate our methods on 3 lexical semantic tasks across 7 datasets and show that our sense-specific VSMs effectively integrate knowledge from the ontology with distributional statistics. Empirically, this results in consistently and significantly better performance over baselines in most cases. In the more marginal cases, analysis reveals that our performance is a result of the deficient structure of the ontology. We discuss and compare the two different approaches from the perspectives of performance, generalizability, flexibility and computational efficiency. Finally, we qualitatively analyze the vectors and show that they indeed capture sense-specific semantics.

2 Unified Symbolic and Distributional Semantics

In this section, we present our two techniques for inferring sense-specific vectors grounded in an ontology. We begin with notation. Let $W = \{w_1, ..., w_n\}$ be a set of word types, and $W_s = \{s_{ij} \mid \forall w_i \in W\ ,\ 1 \leq j \leq k_i\}$ a set of senses, with k_i the number of senses of w_i. Moreover, let $\Omega = (T_\Omega, E_\Omega)$ be an ontology represented by an undirected graph. The vertices $T_\Omega = \{t_{ij} | \ \forall\ s_{ij} \in W_s\}$ correspond to the word senses in the set W_s, while the edges $E_\Omega = \{e^r_{ij-i'j'}\}$ connect some subset of word sense pairs $(s_{ij}, s_{i'j'})$ by semantic relation r[1].

2.1 Retrofitting Vectors to an Ontology

Our first technique assumes that we already have a vector space embedding of a vocabulary $\hat{U} = \{\hat{u}_i | \ \forall\ w_i \in W\}$. We wish to infer vectors $V = \{v_{ij} | \ \forall\ s_{ij} \in W_s\}$ for word senses that are maximally consistent with both $\hat{U}$ and Ω, by some notion of consistency. We formalize this notion as MAP inference in a Markov network (MN).

The MN we propose contains variables for every vector in $\hat{U}$ and V. These variables are connected to one another by dependencies as follows. Variables for vectors v_{ij} and $v_{i'j'}$ are connected iff there exists an edge $e^r_{ij-i'j'} \in E_\Omega$ connecting their respective word senses in the ontology. Furthermore, vectors $\hat{u}_i$ for the word types w_i are each connected to all the vectors v_{ij} of the different senses s_{ij} of w_i. If w_i is not contained in the ontology, we assume it has a single unconnected sense and set it's only sense vector v_{i1} to it's empirical estimate $\hat{u}_i$.

The structure of this MN is illustrated in Figure 1, where the neighborhood of the ambiguous word "bank" is presented as a factor graph.

We set each pairwise clique potential to be of the form $\exp(a\|u - v\|^2)$ between neighboring nodes. Here u and v are the vectors corresponding to these nodes, and a is a weight controlling the strength of the relation between them. We use the Euclidean norm instead of a distance based on cosine similarity because it is more convenient from an optimization perspective.

Our inference problem is to find the MAP estimate of the vectors V, given $\hat{U}$, which may be stated

[1]For example there might be a "synonym" edge between the word senses "cat(1)" and "feline(1)".

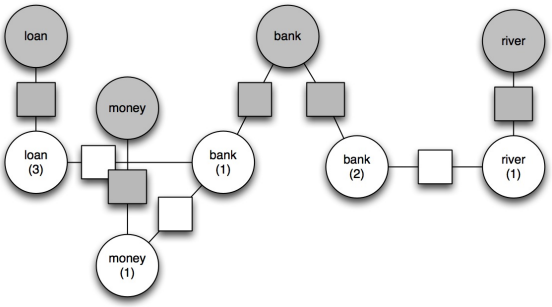

Figure 1: A factor graph depicting the retrofitting model in the neighborhood of the word "bank". Observed variables corresponding to word types are shaded in grey, while latent variables for word senses are in white.

as follows:

$$C(V) = \arg\min_{V} \sum_{i-ij} \alpha \|\hat{u}_i - v_{ij}\|^2 + \sum_{ij-i'j'} \beta_r \|v_{ij} - v_{i'j'}\|^2 \quad (1)$$

Here α is the sense-agnostic weight and β_r are relation-specific weights for different semantic relations. This objective encourages vectors of neighboring nodes in the MN to pull closer together, leveraging the tension between sense-agnostic neighbors (the first summation term) and ontological neighbors (the second summation term). This allows the different neighborhoods of each sense-specific vector to tease it apart from its sense-agnostic vector.

Taking the partial derivative of the objective in equation 1 with respect to vector v_{ij} and setting to zero gives the following solution:

$$v_{ij} = \frac{\alpha\hat{u}_i + \sum_{i'j' \in \mathcal{N}_{ij}} \beta_r v_{i'j'}}{\alpha + \sum_{i'j' \in \mathcal{N}_{ij}} \beta_r} \quad (2)$$

where $\mathcal{N}_{ij}$ denotes the set of neighbors of ij. Thus, the MAP sense-specific vector is an α-weighted combination of its sense-agnostic vector and the β_r-weighted sense-specific vectors in its ontological neighborhood.

We use coordinate descent to iteratively update the variables V using equation 2. The optimization problem in equation 1 is convex, and we normally converge to a numerically satisfactory stationary point within 10 to 15 iterations. This procedure is summarized in Algorithm 1. The generality of this algorithm allows it to be applicable to any VSM as a computationally attractive post-processing step.

Algorithm 1 Outputs a sense-specific VSM, given a sense-agnostic VSM and ontology

```
function RETROFIT(Û, Ω)
    V^(0) ← {v_ij^(0) = û_i | ∀ s_ij ∈ W_s}
    while ||v_ij^(t) − v_ij^(t−1)|| ≥ ε ∀i, j do
        for t_ij ∈ T_Ω do
            v_ij^(t+1) ← update using equation 2
        end for
    end while
    return V^(t)
end function
```

An implementation of this technique is available at `https://github.com/sjauhar/SenseRetrofit`.

2.2 Adapting Predictive Models with Latent Variables and Structured Regularizers

Many successful techniques for semantic representation learning are formulated as models where the desired embeddings are parameters that are learnt to maximize the likelihood of a corpus (Collobert and Weston, 2008; Mnih and Teh, 2012; Mikolov et al., 2013a). In our second approach we extend an existing probability model by adding latent variables representing the senses, and we use a structured prior based on the topology of the ontology to ground the sense embeddings. Formally, we assume a corpus $D = \{(w_1, c_1), \ldots, (w_N, c_N)\}$ of pairs of target and context words, and the ontology Ω, and we wish to infer sense-specific vectors $V = \{v_{ij} | \forall\ s_{ij} \in W_s\}$.

Consider a model with parameters θ ($V \in \theta$) that factorizes the probability over the corpus as $\prod_{(w_i,c_i)\in D} p(w_i, c_i; \theta)$. We propose to extend such a model to learn ontologically grounded sense vectors by presenting a general class of objectives of the following form:

$$C(\theta) = \arg\max_{\theta} \sum_{(w_i,c_i)\in D} \log(\sum_{s_{ij}} p(w_i, c_i, s_{ij}; \theta)) + \log p_\Omega(\theta) \quad (3)$$

This objective introduces latent variables s_{ij} for senses and adds a structured regularizer $p_\Omega(\theta)$

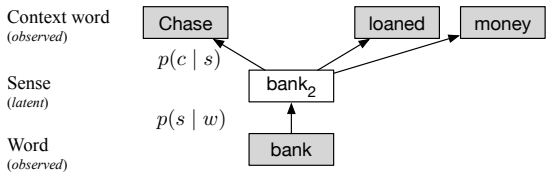

Figure 2: The generative process associated with the skip-gram model, modified to account for latent senses. Here, the context of the ambiguous word "bank" is generated from the selection of a specific latent sense.

that grounds the vectors V in an ontology. This form permits flexibility in the definition of both $p(w_i, c_i, s_{ij}; \theta)$ and $p_\Omega(\theta)$ allowing for a general yet powerful framework for adapting MLE models.

In what follows we show that the popular skip-gram model (Mikolov et al., 2013a) can be adapted to generate ontologically grounded sense vectors. The classic skip-gram model uses a set of parameters $\theta = (U, V)$, with $U = \{u_i \mid \forall c_i \in W\}$ and $V = \{v_i \mid \forall w_i \in W\}$ being sets of vectors for context and target words respectively. The generative story of the skip-gram model involves generating the context word c_i conditioned on an observed word w_i. The conditional probability is defined to be $p(c_i \mid w_i; \theta) = \frac{\exp(u_i \cdot v_i)}{\sum_{c_{i'} \in W} \exp(u_{i'} \cdot v_i)}$.

We modify the generative story of the skip-gram model to account for latent sense variables by first selecting a latent word sense s_{ij} conditional on the observed word w_i, then generating the context word c_i from the sense distinguished word s_{ij}. This process is illustrated in Figure 2. The factorization $p(c_i \mid w_i; \theta) = \sum_{s_{ij}} p(c_i \mid s_{ij}; \theta) \times p(s_{ij} \mid w_i; \theta)$ follows from the chain rule since senses are word-specific. To parameterize this distribution, we define a new set of model parameters $\theta = (U, V, \Pi)$, where U remains identical to the original skip-gram, $V = \{v_{ij} \mid \forall s_{ij} \in W_s\}$ are a set of vectors for word senses, and Π are the context-independent sense proportions $\pi_{ij} = p(s_{ij} \mid w_i)$. We use a Dirichlet prior over the multinomial distributions π_i for every w_i, with a shared concentration parameter λ.

We define the ontological prior on vectors as $p_\Omega(\theta) \propto \exp(-\gamma \sum_{ij-i'j'} \beta_r \|v_{ij} - v_{i'j'}\|^2)$, where γ controls the strength of the prior. We note the similarity to the retrofitting objective in equation 1, except with $\alpha = 0$. This leads to the following realization of the objective in equation 3:

$$C(\theta) = \arg\max_\theta \sum_{(w_i, c_i) \in D} \log \Big(\sum_{s_{ij}} p(c_i \mid s_{ij}; \theta) \times p(s_{ij} \mid w_i; \theta) \Big) - \gamma \sum_{ij-i'j'} \beta_r \|v_{ij} - v_{i'j'}\|^2 \quad (4)$$

This objective can be optimized using EM, for the latent variables, and with lazy updates (Carpenter, 2008) every k words to account for the prior regularizer. However, since we are primarily interested in learning good vector representations, and we want to learn efficiently from large datasets, we make the following simplifications. First, we perform "hard" EM, selecting the most likely sense at each position rather than using the full posterior over senses. Also, given that the structured regularizer $p_\Omega(\theta)$ is essentially the retrofitting objective in equation 1, we run retrofitting periodically every k words (with $\alpha = 0$ in equation 2) instead of lazy updates.[2]

The following decision rule is used in the "hard" E-step:

$$s_{ij} = \arg\max_{s_{ij}} p(c_i \mid s_{ij}; \theta^{(t)}) \pi_{ij}^{(t)} \quad (5)$$

In the M-step we use Variational Bayes to update Π with:

$$\pi_{ij}^{(t+1)} \propto \frac{\exp\left(\psi\left(\tilde{c}(w_i, s_{ij}) + \lambda \pi_{ij}^{(0)}\right)\right)}{\exp\left(\psi\left(\tilde{c}(w_i) + \lambda\right)\right)} \quad (6)$$

where $\tilde{c}(\cdot)$ is the online expected count and $\psi(\cdot)$ is the digamma function. This approach is motivated by Johnson (2007) who found that naive EM leads to poor results, while Variational Bayes is consistently better and promotes faster convergence of the likelihood function. To update the parameters U and V, we use negative sampling (Mikolov et al., 2013a) which is an efficient approximation to the original skip-gram objective. Negative sampling attempts to distinguish between true word pairs in the data, relative to noise. Stochastic gradient descent on the following equation is used to update the model pa-

[2] We find this gives slightly better performance.

rameters U and V:

$$\mathcal{L} = \log \sigma(u_i \cdot v_{ij}) + \sum_{\substack{j' \\ j' \neq j}} \log \sigma(-u_i \cdot v_{ij'}) + \sum_{m} \mathbb{E}_{c_{i'} \sim P_n(c)} \left[\log \sigma(-u_{i'} \cdot v_{ij}) \right] \quad (7)$$

Here $\sigma(\cdot)$ is the sigmoid function, $P_n(c)$ is a noise distribution computed over unigrams and m is the negative sampling parameter. This is almost exactly the same as negative sampling proposed for the original skip-gram model. The only change is that we additionally take a negative gradient step with respect to all the senses that were not selected in the hard E-step. We summarize the training procedure for the adapted skip-gram model in Algorithm 2.

Algorithm 2 Outputs a sense-specific VSM, given a corpus and an ontology

function SENSEEM(D, Ω)
  $\theta^{(0)} \leftarrow$ initialize
  for $(w_i, c_i) \in D$ **do**
    if period > k **then**
      RETROFIT($\theta^{(t)}, \Omega$)
    end if
    (Hard) E-step:
    $s_{ij} \leftarrow$ find argmax using equation 5
    M-step:
    $\Pi^{(t+1)} \leftarrow$ update using equation 6
    $U^{(t+1)}, V^{(t+1)} \leftarrow$ update using equation 7
  end for
  return $\theta^{(t)}$
end function

3 Evaluation

In this section we detail experimental results on 3 lexical semantics tasks across 8 different datasets. We begin by detailing the training and setup for our experiments.

3.1 Resources, Data and Training

We use WordNet (Miller, 1995) as the sense repository and ontology in all our experiments. WordNet is a large, hand-annotated ontology of English composed of 117,000 clusters of senses, or "synsets" that are related to one another through semantic relations such as hypernymy and hyponymy. Each synset additionally comprises a list of sense specific lemmas which we use to form the nodes in our graph. There are 206,949 such sense specific lemmas, which we connect with synonym, hypernym and hyponym[3] relations for a total of 488,432 edges.

To show the applicability of our techniques to different VSMs we experiment with two different kinds of base vectors.

Global Context Vectors (GC) (Huang et al., 2012): These word vectors were trained using a neural network which not only uses local context but also defines global features at the document level to further enhance the VSM. We distinguish three variants: the original single-sense vectors (SINGLE), a multi-prototype variant (MULTI), – both are available as pre-trained vectors for download[4] – and a sense-based version obtained by running retrofitting on the original vectors (RETRO).

Skip-gram Vectors (SG) (Mikolov et al., 2013a): We use the word vector tool `Word2Vec`[5] to train skip-gram vectors. We define 6 variants: a single-sense version (SINGLE), two multi-sense variants that were trained by first sense disambiguating the entire corpus using WSD tools, – one unsupervised (Pedersen and Kolhatkar, 2009) (WSD) and the other supervised (Zhong and Ng, 2010) (IMS) – a retrofitted version obtained from the single-sense vectors (RETRO), an EM implementation of the skip-gram model with the structured regularizer as described in section 2.2 (EM+RETRO), and the same EM technique but ignoring the ontology (EM). All models were trained on publicly available WMT-2011[6] English monolingual data. This corpus of 355 million words, although adequate in size, is smaller than typically used billion word corpora. We use this corpus because the WSD baseline involves preprocessing the corpus with sense disambiguation, which is slow enough that running it on corpora orders of magnitude larger was infeasible.

Retrofitted variants of vectors (RETRO) are trained using the procedure described in algorithm 1. We set the convergence criteria to $\epsilon = 0.01$ with a maximum number of iterations of 10. The weights

[3] We treat edges as undirected, so hypernymy and hyponymy are collapsed and unified in our representation schema.

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

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

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

in the update equation 2 are set heuristically: the sense agnostic weight α is 1.0, and relations-specific weights β_r are 1.0 for synonyms and 0.5 for hypernyms and hyponyms. EM+RETRO vectors are the exception where we use a weight of $\alpha = 0.0$ instead, as required by the derivation in section 2.2.

For skip-gram vectors (SG) we use the following standard settings, and do not tune any of the values. We filter all words with frequency < 5, and pre-normalize the corpus to replace all numeric tokens with a placeholder. We set the dimensionality of the vectors to 80, and the window size to 10 (5 context words to either side of a target). The learning rate is set to an initial value of 0.025 and diminished linearly throughout training. The negative sampling parameter is set to 5. Additionally for the EM variants (section 2.2) we set the Dirichlet concentration parameter λ to 1000. We use 5 abstract senses for the EM vectors, and initialize the priors uniformly. For EM+RETRO, WordNet dictates the number of senses; also when available WordNet lemma counts are used to initialize the priors. Finally, we set the retrofitting period k to 50 million words.

3.2 Experimental Results

We evaluate our models on 3 kinds of lexical semantic tasks: similarity scoring, synonym selection, and similarity scoring in context.

Similarity Scoring: This task involves using a semantic model to assign a score to pairs of words. We use the following 4 standard datasets in this evaluation: WS-353 (Finkelstein et al., 2002), RG-65 (Rubenstein and Goodenough, 1965), MC-30 (Miller and Charles, 1991) and MEN-3k (Bruni et al., 2014). Each dataset consists of pairs of words along with an averaged similarity score obtained from several human annotators. For example an item in the WS-353 dataset is "book, paper → 7.46". We use standard cosine similarity to assign a score to word pairs in single-sense VSMs, and the following average similarity score to multi-sense variants, as proposed by Reisinger and Mooney (2010):

$$avgSim(w_i, w_{i'}) = \frac{1}{k_i k_j} \sum_{j,j'} cos(v_{ij}, v_{i'j'}) \quad (8)$$

The output of systems is evaluated against the gold standard using Spearman's rank correlation coefficient.

Synonym Selection: In this task, VSMs are used to select the semantically closest word to a target from a list of candidates. We use the following 3 standard datasets in this evaluation: ESL-50 (Turney, 2001), RD-300 (Jarmasz and Szpakowicz, 2004) and TOEFL-80 (Landauer and Dumais, 1997). These datasets consist of a list of target words that appear with several candidate lexical items. An example from the TOEFL dataset is "rug → sofa, ottoman, carpet, hallway", with "carpet" being the most synonym-like candidate to the target. We begin by scoring all pairs composed of the target and one of the candidates. We use cosine similarity for single-sense VSMs, and max similarity for multi-sense models[7]:

$$maxSim(w_i, w_{i'}) = \max_{j,j'} cos(v_{ij}, v_{i'j'}) \quad (9)$$

These scores are then sorted in descending order, with the top-ranking score yielding the semantically closest candidate to the target. Systems are evaluated on the basis of their accuracy at discriminating the top-ranked candidate.

The results for similarity scoring and synonym selection are presented in table 1. On both tasks and on all datasets, with the partial exception of WS-353 and MEN-3k, our vectors (RETRO & EM+RETRO) consistently yield better results than other VSMs. Notably, both our techniques perform better than preprocessing a corpus with WSD information in unsupervised or supervised fashion (SG-WSD & SG-IMS). Simple EM without an ontological prior to ground the vectors (SG-EM) also performs poorly.

We investigated the observed drop in performance on WS-353 and found that this dataset consists of two parts: a set of *similar* word pairs (e.g. "tiger" and "cat") and another set of *related* word pairs (e.g. "weather" and "forecast"). The synonym, hypernym and hyponym relations we use tend to encourage *similarity* to the detriment of *relatedness*.

We ran an auxiliary experiment to show this. SG-EM+RETRO training also learns vectors for context words – which can be thought of as a proxy for relatedness. Using this VSM we scored a word pair by the average similarity of all the sense vectors of

[7] Here we are specifically looking for synonyms, so the max makes more sense than taking an average.

		Word Similarity (ρ)				Synonym Selection (%)		
		WS-353	RG-65	MC-30	MEN-3k	ESL-50	RD-300	TOEFL-80
GC	SINGLE	**0.623**	0.629	0.657	0.314	47.73	45.07	60.87
	MULTI	0.535	0.510	0.309	0.359	27.27	47.89	52.17
	RETRO	0.543	**0.661**	**0.714**	**0.528**	**63.64**	**66.20**	**71.01**
SG	SINGLE	**0.639**	0.546	0.627	**0.646**	52.08	55.66	66.67
	EM	0.194	0.278	0.167	0.228	27.08	33.96	40.00
	WSD	0.481	0.298	0.396	0.175	16.67	49.06	42.67
	IMS	0.549	0.579	0.606	0.591	41.67	53.77	66.67
	RETRO	0.552	0.673	0.705	0.560	56.25	65.09	**73.33**
	EM+RETRO	0.321	**0.734**	**0.758**	0.428	**62.22**	**66.67**	68.63

Table 1: Similarity scoring and synonym selection in English across several datasets involving different VSMs. Higher scores are better; best scores within each category are in bold. In most cases our models consistently and significantly outperform the other VSMs.

one word to the context vector of the other word, averaged over both words. With this scoring function the correlation ρ jumped from 0.321 to 0.493. While still not as good as some of the other VSMs, it should be noted that this scoring function negatively influences the *similar* word pairs in the dataset.

The MEN-3k dataset is crowd-sourced and contains much diversity, with word pairs evidencing similarity as well as relatedness. However, we aren't sure why the performance for GC-RETRO improves greatly over GC-SINGLE for this dataset, while that of SG-RETRO and SG-RETRO+EM drops in relation to SG-SINGLE.

Similarity Scoring in Context: As outlined by Reisinger and Mooney (2010), multi-sense VSMs can be used to consider context when computing similarity between words. We use the SCWS dataset (Huang et al., 2012) in these experiments. This dataset is similar to the similarity scoring datasets, except that they additionally are presented in context. For example an item involving the words "bank" and "money", gives the words in their respective contexts, "along the east **bank** of the Des Moines River" and "the basis of all **money** laundering" with a low averaged similarity score of 2.5 (on a scale of 1.0 to 10.0). Following Reisinger and Mooney (2010) we use the following function to assign a score to word pairs in their respective contexts, given a multi-sense VSM:

$$avgSimC(w_i, c_i, w_{i'}, c_{i'}) = \sum_{j,j'} p(s_{ij}|c_i, w_i)p(s_{i'j'}|c_{i'}, w_{i'})cos(v_{ij}, v_{i'j'}) \quad (10)$$

Vectors	SCWS (ρ)
SG-WSD	0.343
SG-IMS	0.528
SG-RETRO	0.417
GC-RETRO	0.420
SG-EM	0.613
SG-EM+RETRO	0.587
GC-MULTI	**0.657**

Table 2: Contextual word similarity in English. Higher scores are better.

As with similarity scoring, the output of systems is evaluated against gold standard using Spearman's rank correlation coefficient.

The results are presented in table 2. Preprocessing a corpus with WSD information in an unsupervised fashion (SG-WSD) yields poor results. In comparison, the retrofitted vectors (SG-RETRO & GC-RETRO) already perform better, even though they do not have access to context vectors, and thus do not take contextual information into account. Supervised sense vectors (SG-IMS) are also competent, scoring better than both retrofitting techniques. Our EM vectors (SG-EM & SG-EM+RETRO) yield even better results and are able to capitalize on contextual information, however they still fall short of the pretrained GC-MULTI vectors. We were surprised that SG-EM+RETRO actually performed worse than SG-EM, given how poorly SG-EM performed in the other evaluations. However, an analysis again revealed that this was due to the kind of similarity encouraged by WordNet rather than an inability of the model to learn useful vectors. The SCWS dataset, in addition to containing related

Method	CPU Time
RETRO	~20 secs
EM+RETRO	~4 hours
IMS	~3 days
WSD	~1 year

Table 3: Training time associated with different methods of generating sense-specific VSMs.

words – which we showed, hurt our performance on WS-353 – also contains word pairs with different POS tags. WordNet synonymy, hypernymy and hyponymy relations are exclusively defined between lemmas of the same POS tag, which adversely affects performance further.

3.3 Discussion

While both our approaches are capable of integrating ontological information into VSMs, an important question is which one should be preferred? From an empirical point of view, the EM+RETRO framework yields better performance than RETRO across most of our semantic evaluations. Additionally EM+RETRO is more powerful, allowing to adapt more expressive models that can jointly learn other useful parameters – such as context vectors in the case of skip-gram. However, RETRO is far more generalizable, allowing it to be used for *any* VSM, not just predictive MLE models, and is also empirically competitive. Another consideration is computational efficiency, which is summarized in table 3.

Not only is RETRO much faster, but it scales linearly with respect to the vocabulary size, unlike EM+RETRO, WSD, and IMS which are dependent on the input training corpus. Nevertheless, both our techniques are empirically superior as well as computationally more efficient than both unsupervised and supervised word-sense disambiguation paradigms.

Both our approaches are sensitive to the structure of the ontology. Therefore, an important consideration is the relations we use and the weights we associate with them. In our experiments we selected the simplest set of relations and assigned weights heuristically, showing that our methods can effectively integrate ontological information into VSMs. A more exhaustive selection procedure with weight tuning on held-out data would almost certainly lead to better performance on our evaluation suite.

3.4 Qualitative Analysis

We qualitatively attempt to address the question of whether the vectors are truly sense specific. In table 4 we present the three most similar words of an ambiguous lexical item in a standard VSM (SG-SINGLE) in comparison with the three most similar words of different lemma senses of the same lexical item in grounded sense VSMs (SG-RETRO & SG-EM+RETRO).

Word or Sense	Top 3 Most Similar
hanging	hung dangled hangs
hanging (suspending)	shoring support suspension
hanging (decoration)	tapestry braid smock
climber	climbers skier Loretan
climber (sportsman)	lifter swinger sharpshooter
climber (vine)	woodbine brier kiwi

Table 4: The top 3 most similar words for two polysemous types. Single sense VSMs capture the most frequent sense. Our techniques effectively separates out the different senses of words, and are grounded in WordNet.

The sense-agnostic VSMs tend to capture only the most frequent sense of a lexical item. On the other hand, the disambiguated vectors capture sense specificity of even less frequent senses successfully. This is probably due to the nature of WordNet where the nearest neighbors of the words in question are in fact these rare words. A careful tuning of weights will likely optimize the trade-off between ontologically rare neighbors and distributionally common words.

In our analyses, we noticed that lemma senses that had many neighbors (i.e. synonyms, hypernyms and hyponyms), tended to have more clearly sense specific vectors. This is expected, since it is these neighborhoods that disambiguate and help to distinguish the vectors from their single sense embeddings.

4 Related Work

Since Reisinger and Mooney (2010) first proposed a simple context clustering technique to generate multi-prototype VSMs, a number of related efforts have worked on adaptations and improvements relying on the same clustering principle. Huang et al. (2012) train their vectors with a neural network and additionally take global context into account. Neelakantan et al. (2014) extend the popular skip-gram model (Mikolov et al., 2013a) in a non-parametric

fashion to allow for different number of senses for words. Guo et al. (2014) exploit bilingual alignments to perform better context clustering during training. Tian et al. (2014) propose a probabilistic extension to skip-gram that treats the different prototypes as latent variables. This is similar to our second EM training framework, and turns out to be a special case of our general model. In all these papers, however, the multiple senses remain abstract and are not grounded in an ontology.

Conceptually, our work is also similar to Yu and Dredze (2014) and Faruqui et al. (2014), who treat lexicons such as the paraphrase database (PPDB) (Ganitkevitch et al., 2013) or WordNet (Miller, 1995) as an auxiliary thesaurus to improve VSMs. However, they do not model senses in any way. Pilehvar et al. (2013) do model senses from an ontology by performing random-walks on the Wordnet graph, however their approach does not take distributional information from VSMs into account.

Thus, to the best of our knowledge, our work presents the first attempt at producing sense grounded VSMs that are symbolically tied to lexical ontologies. From a modelling point of view, it is also the first to outline a unified, principled and extensible framework that effectively combines the symbolic and distributional paradigms of semantics.

Both our models leverage the graph structure of ontologies to effectively ground the senses of a VSM. This ties into previous research (Das and Smith, 2011; Das and Petrov, 2011) that propagates information through a factor graph to perform tasks such as frame-semantic parsing and POS-tagging across languages. More generally, this approach can be viewed from the perspective of semi-supervised learning, with an optimization over a graph loss function defined on smoothness properties (Corduneanu and Jaakkola, 2002; Zhu et al., 2003; Subramanya and Bilmes, 2009).

Related to the problem of polysemy is the issue of different shades of meaning a word assumes based on context. The space of research on this topic can be divided into three broad categories: models for computing contextual lexical semantics based on composition (Mitchell and Lapata, 2008; Erk and Padó, 2008; Thater et al., 2011), models that use fuzzy exemplar-based contexts without composing them (Erk and Padó, 2010; Reddy et al., 2011), and models that propose latent variable techniques (Dinu and Lapata, 2010; Séaghdha and Korhonen, 2011; Van de Cruys et al., 2011). Our work, which tackles the stronger form of lexical ambiguity in polysemy falls into the latter two of three categories.

5 Conclusion and Future Work

We have presented two general and flexible approaches to producing sense-specific VSMs grounded in an ontology. The first technique is applicable to any VSM as an efficient post-processing step while the second provides a framework to integrate ontological information with existing MLE-based predictive models. We presented an evaluation of 3 semantic tasks on 7 datasets. Our results show that our proposed methods are effectively able to capture the different senses in an ontology. In most cases this results in significant improvements over baselines. We have also discussed the trade-offs between the two techniques from several different perspectives. Finally, we have presented a qualitative analysis investigating the nature of the sense-specific vectors, and shown that they capture the semantics of different senses.

Our findings suggest several avenues for future research. We propose to use sense-specific vectors as features in downstream applications such a Word Sense Disambiguation. Our current approach assumes a fixed ontology, but we hope to explore a more bi-directional relationship between ontology and VSM in future work. In particularly we envisage simultaneously incrementing ontologies with structure learning in addition to improving VSMs. We also hope to extend our research to the multi-lingual domain. We are particularly excited by the idea of using multi-lingual WordNets to learn sense specific semantic vectors that generalize across languages.

Acknowledgments

The authors would like to thank Manaal Faruqui, Jesse Dodge and Noah Smith for their insight and feedback. Thanks also go to the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work was supported in part by the following grants: NSF grant IIS-1143703, NSF award IIS-1147810, DARPA grant FA87501220342.

References

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research (JAIR)*, 49:1–47.

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.

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.

Adrian Corduneanu and Tommi Jaakkola. 2002. On information regularization. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 151–158. Morgan Kaufmann Publishers Inc.

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*.

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*.

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1162–1172. Association for Computational Linguistics.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 897–906, Stroudsburg, PA, USA. Association for Computational Linguistics.

Katrin Erk and Sebastian Padó. 2010. Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 92–97. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. In *ACM Transactions on Information Systems*, volume 20, pages 116–131, January.

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.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning sense-specific word embeddings by exploiting bilingual resources. In *Proceedings of COLING*, pages 497–507.

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.

Mario Jarmasz and Stan Szpakowicz. 2004. Roget's thesaurus and semantic similarity. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 2003:111.

Mark Johnson. 2007. Why doesn't em find good hmm pos-taggers? In *EMNLP-CoNLL*, pages 296–305. Citeseer.

Adam Kilgarriff. 1997. I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

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.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

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 Miller and Walter Charles. 1991. Contextual correlates of semantic similarity. In *Language and Cognitive Processes*, pages 1–28.

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

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244.

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.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of EMNLP*.

Ted Pedersen and Varada Kolhatkar. 2009. Wordnet:: Senserelate:: Allwords: a broad coverage word sense tagger that maximizes semantic relatedness. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics, companion volume: Demonstration session*, pages 17–20. Association for Computational Linguistics.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *ACL (1)*, pages 1341–1351.

Siva Reddy, Ioannis P Klapaftis, Diana McCarthy, and Suresh Manandhar. 2011. Dynamic and static prototype vectors for semantic composition. In *IJCNLP*, pages 705–713.

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.

Diarmuid Ó Séaghdha and Anna Korhonen. 2011. Probabilistic models of similarity in syntactic context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1047–1057, Stroudsburg, PA, USA. Association for Computational Linguistics.

Amarnag Subramanya and Jeff A Bilmes. 2009. Entropic graph regularization in non-parametric semi-supervised classification. In *NIPS*, pages 1803–1811.

Stefanie Tellex, Boris Katz, Jimmy J. Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR*, pages 41–47.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word meaning in context: A simple and effective vector model. In *IJCNLP*, pages 1134–1143.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING*, pages 151–160.

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.

Peter D Turney. 2013. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *Transactions of the Association for Computational Linguistics*, 1:353–366.

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2011. Latent vector weighting for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1012–1022. Association for Computational Linguistics.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83. Association for Computational Linguistics.

Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919.