Learning Semantic Word Embeddings based on Ordinal Knowledge Constraints

Quan Liu[†] and **Hui Jiang**[‡] and **Si Wei**[§] and **Zhen-Hua Ling**[†] and **Yu Hu**[§]
[†] National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, China
[‡] Department of Electrical Engineering and Computer Science, York University, Canada
[§] iFLYTEK Research, Hefei, China
emails: quanliu@mail.ustc.edu.cn, hj@cse.yorku.ca,
siwei@iflytek.com, zhling@ustc.edu.cn, yuhu@iflytek.com

Abstract

In this paper, we propose a general framework to incorporate semantic knowledge into the popular data-driven learning process of word embeddings to improve the quality of them. Under this framework, we represent semantic knowledge as many ordinal ranking inequalities and formulate the learning of semantic word embeddings (SWE) as a constrained optimization problem, where the data-derived objective function is optimized subject to all ordinal knowledge inequality constraints extracted from available knowledge resources such as Thesaurus and WordNet. We have demonstrated that this constrained optimization problem can be efficiently solved by the stochastic gradient descent (SGD) algorithm, even for a large number of inequality constraints. Experimental results on four standard NLP tasks, including word similarity measure, sentence completion, name entity recognition, and the TOEFL synonym selection, have all demonstrated that the quality of learned word vectors can be significantly improved after semantic knowledge is incorporated as inequality constraints during the learning process of word embeddings.

1 Introduction

Distributed word representation (i.e., word embedding) is a technique that represents words as continuous vectors, which is an important research topic in natural language processing (NLP) (Hinton et al., 1986; Turney et al., 2010). In recent years, it has been widely used in various NLP tasks, including neural language model (Bengio et al., 2003; Schwenk, 2007), sequence labelling tasks (Collobert and Weston, 2008; Collobert et al., 2011), machine translation (Devlin et al., 2014; Sutskever et al., 2014), and antonym selection (Chen et al., 2015). Typically, word vectors are learned based on *the distributional hypothesis* (Harris, 1954; Miller and Charles, 1991), which assumes that words with a similar context tend to have a similar meaning. Under this hypothesis, various models, such as the skip-gram model (Mikolov et al., 2013a; Mikolov et al., 2013b; Levy and Goldberg, 2014) and GloVe model (Pennington et al., 2014), have been proposed to leverage the context of each word in large corpora to learn word embeddings. These methods can efficiently estimate the co-occurrence statistics to model contextual distributions from very large text corpora and they have been demonstrated to be quite effective in a number of NLP tasks. However, they still suffer from some major limitations. In particular, these corpus-based methods usually fail to capture the precise meanings for many words. For example, some semantically related but dissimilar words may have similar contexts, such as synonyms and antonyms. As a result, these corpus-based methods may lead to some antonymous word vectors being located much closer in the learned embedding space than many synonymous words. Moreover, as word representations are mainly learned based on the co-occurrence information, the learned word embeddings do not capture the accurate relationship between two semantically similar words if either one appears less frequently in the corpus.

To address these issues, some recent work has been proposed to incorporate prior lexical knowledge (WordNet, PPDB, etc.) or knowledge graph (Freebase, etc.) into word representations. Such knowledge enhanced word embedding methods have achieved considerable improvements on various natural language processing tasks, like (Yu and Dredze, 2014; Bian et al., 2014; Xu et al., 2014). These methods attempt to increase the semantic similarities between words belonging to one semantic category or to explicitly model the semantic relationships between different words. For example, Yu and Dredze (2014) have proposed a new learning objective function to enhance word embeddings by combining neural models and a prior knowledge measure from semantic resources. Bian et. al (2014) have recently proposed to leverage morphological, syntactic, and semantic knowledge to improve the learning of word embeddings. Besides, a novel framework has been proposed in (Xu et al., 2014) to take advantage of both relational and categorical knowledge to learn high-quality word representations, where two regularization functions are used to model the relational and categorical knowledge respectively. More recently, a retrofitting technique has been introduced in (Faruqui et al., 2014) to improve semantic vectors by leveraging lexicon-derived relational information in a post-processing stage.

In this paper, we propose a new and flexible method to incorporate semantic knowledge into the corpus-based learning of word embeddings. In our approach, we propose to represent semantic knowledge as many word ordinal ranking inequalities. Furthermore, these inequalities are cast as semantic constraints in the optimization process to learn semantically sensible word embeddings. The proposed method has several advantages. Firstly, many different types of semantic knowledge can all be represented as a number of such ranking inequalities, such as synonym-antonym, hyponym-hypernym and etc. Secondly, these inequalities can be easily extracted from many existing knowledge resources, such as Thesaurus, WordNet (Miller, 1995) and knowledge graphs. Moreover, the ranking inequalities can also be manually generated by human annotation because ranking orders is much easier for human annotators than assigning specific scores. Next, we present a flexible learning framework to learn distributed word representation based on the *ordinal semantic knowledge*. By solving a constrained optimization problem using the efficient stochastic gradient descent algorithm, we can obtain semantic word embedding enhanced by the ordinal knowledge constraints. Experiments on four popular natural language processing tasks, including word similarity, sentence completion, name entity recognition and synonym selection, have all demonstrated that the proposed method can learn good semantically sensible word embeddings.

2 Representing Knowledge By Ranking

Many types of lexical semantic knowledge can be quantitatively represented by a large number of ranking inequalities such as:

$$\mathrm{similarity}(w_i, w_j) > \mathrm{similarity}(w_i, w_k) \quad (1)$$

where w_i, w_j and w_k denote any three words in vocabulary. For example, eq.(1) holds if w_j is a synonym of w_i and w_k is an antonym of w_i. In general, the similarity between a word and its synonymous word should be larger than the similarity between the word and its antonymous word. Moreover, a particular word should be more similar to the words belonging to the same semantic category as this word than other words belonging to a different category. Besides, eq.(1) holds if w_i and w_j have shorter distance in a semantic hierarchy than w_i and w_k do in the same hierarchy (Leacock and Chodorow, 1998; Jurafsky and Martin, 2000).

Equivalently, each of the above similarity inequalities may be represented as the following constraint in the embedding space:

$$\mathrm{sim}(\mathbf{w}_i^{(1)}, \mathbf{w}_j^{(1)}) > \mathrm{sim}(\mathbf{w}_i^{(1)}, \mathbf{w}_k^{(1)}) \quad (2)$$

where $\mathbf{w}_i^{(1)}$, $\mathbf{w}_j^{(1)}$ and $\mathbf{w}_k^{(1)}$ denote the embedding vectors of the words, w_i, w_j and w_k.

In this paper, we use the following three rules to gather the ordinal semantic knowledge from available lexical knowledge resources, such as Thesaurus and WordNet.

- *Synonym Antonym Rule*: Similarities between a word and its synonymous words are always larger than similarities between the word and its antonymous words. For example, the similarity between *foolish* and *stupid* is expected to be bigger than the similarity between *foolish* and *clever*, i.e., similarity(*foolish*, *stupid*) > similarity(*foolish*, *clever*).

- *Semantic Category Rule*: Similarities of words that belong to the same semantic category would be larger than similarities of words that belong to different categories. This rule refers to the idea of Fisher linear discriminant algorithm in (Fisher, 1936). A semantic category may be defined as a synset in WordNet, a hypernym in a semantic hierarchy, or an entity category in

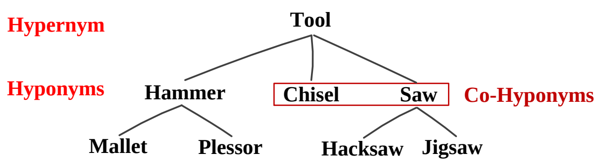

Figure 1: An example of hyponym and hypernym.

knowledge graphs. Figure 1 shows a simple example of the relationship between hyponyms and hypernyms. From there, it is reasonable to assume the following similarity inequality: similarity$(Mallet, Plessor) >$ similarity$(Mallet, Hacksaw)$.

- *Semantic Hierarchy Rule*: Similarities between words that have shorter distances in a semantic hierarchy should be larger than similarities of words that have longer distances. In this work, the semantic hierarchy refers to the hypernym and hyponym structure in WordNet. From Figure 1, this rule may suggest several inequalities like: similarity$(Mallet, Hammer) >$ similarity$(Mallet, Tool)$.

In addition, we may generate many such semantically ranking similarity inequalities by human annotation through crowdsourcing.

3 Semantic Word Embedding

In this section, we first briefly review the conventional skip-gram model (Mikolov et al., 2013b). Next, we study how to incorporate the ordinal similarity inequalities to learn semantic word embeddings.

3.1 The skip-gram model

The skip-gram model is a recently proposed learning framework (Mikolov et al., 2013b; Mikolov et al., 2013a) to learn continuous word vectors from text corpora based on the aforementioned distributional hypothesis, where each word in vocabulary (size of V) is mapped to a continuous embedding space by looking up an embedding matrix $\mathbf{W}^{(1)}$. And $\mathbf{W}^{(1)}$ is learned by maximizing the prediction probability, calculated by another prediction matrix $\mathbf{W}^{(2)}$, of its neighbouring words within a context window.

Given a sequence of training data, denoted as $w_1, w_2, w_3, ..., w_T$ with T words, the skip-gram model aims to maximize the following objective function:

$$\mathcal{Q} = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t) \qquad (3)$$

where c is the size of context windows, w_t denotes the input central word and w_{t+j} for its neighbouring word. The skip-gram model computes the above conditional probability $p(w_{t+j}|w_t)$ using the following softmax function:

$$p(w_{t+j}|w_t) = \frac{\exp(\mathbf{w}_{t+j}^{(2)} \cdot \mathbf{w}_t^{(1)})}{\sum_{k=1}^{V}\exp(\mathbf{w}_k^{(2)} \cdot \mathbf{w}_t^{(1)})} \qquad (4)$$

where $\mathbf{w}_t^{(1)}$ and $\mathbf{w}_k^{(2)}$ denotes row vectors in matrices $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$, corresponding to word w_t and w_k respectively.

The training process of the skip-gram model can be formulated as an optimization problem to maximize the above objective function $\mathcal{Q}$. As in (Mikolov et al., 2013b), this optimization problem is solved by the stochastic gradient descent (SGD) method and the learned embedding matrix $\mathbf{W}^{(1)}$ is used as the word embeddings for all words in vocabulary.

3.2 Semantic Word Embedding (SWE) as Constrained Optimization

Here we consider how to combine the ordinal knowledge representation in section 2 and the skip-gram model in 3.1 to learn semantic word embeddings (SWE).

As shown in section 2, each ranking inequality involves a triplet, (i, j, k), of three words, $\{w_i, w_j, w_k\}$. Assume the ordinal knowledge is represented by a large number of such inequalities, denoted as the inequality set S. For $\forall(i, j, k) \in S$, we have:

$$\begin{aligned} &\text{similarity}(w_i, w_j) > \text{similarity}(w_i, w_k) \\ &\Leftrightarrow \text{sim}(\mathbf{w}_i^{(1)}, \mathbf{w}_j^{(1)}) > \text{sim}(\mathbf{w}_i^{(1)}, \mathbf{w}_k^{(1)}). \end{aligned}$$

For notational simplicity, we denote $s_{ij} = \text{sim}(\mathbf{w}_i^{(1)}, \mathbf{w}_j^{(1)})$ hereafter.

Next, we propose to use the following constrained optimization problem to learn semantic word embeddings (SWE):

$$\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}\} = \arg\max_{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}} \mathcal{Q}(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}) \qquad (5)$$

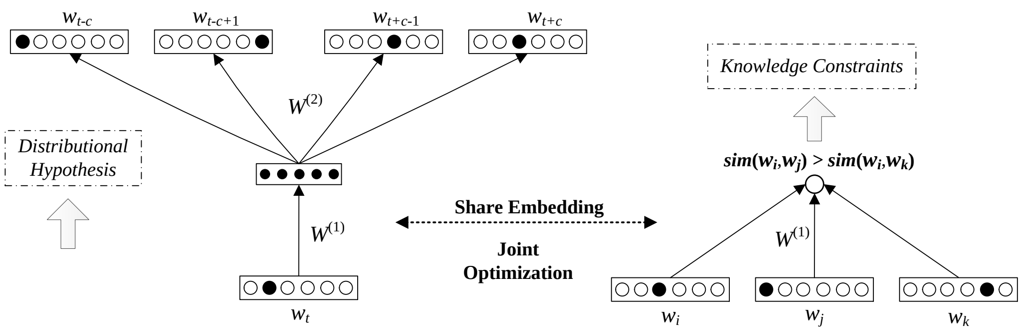

Figure 2: The proposed semantic word embedding (SWE) learning framework (The left part denotes the state-of-the-art skip-gram model; The right part represents the semantic constraints).

subject to

$$s_{ij} > s_{ik} \quad \forall (i,j,k) \in S. \tag{6}$$

In this work, we formulate the above constrained optimization problem into an unconstrained one by casting all the constraints as a penalty term in the objective function. The penalty term can be expressed as follows:

$$\mathcal{D} = \sum_{(i,j,k)\in S} f(i,j,k) \tag{7}$$

where the function $f(\cdot)$ is a normalization function. It can be a sigmoid function like $f(i,j,k) = \sigma(s_{ik} - s_{ij})$ with $\sigma(x) = 1/(1+\exp(-x))$. Alternatively, it may be a hinge loss function like $f(i,j,k) = h(s_{ik} - s_{ij})$ where $h(x) = \max(\delta_0, x)$ with δ_0 denoting a parameter to control the decision margin. In this work, we adopt to use the hinge function to compute the penalty term in eq.(7) and δ_0 is set to be 0 for all experiments.

Finally, the proposed semantic word embedding (SWE) model aims to maximize the following combined objective function:

$$\mathcal{Q}' = \mathcal{Q} - \beta \cdot \mathcal{D} \tag{8}$$

where β is a control parameter to balance the contribution of the penalty term in the optimization process. It balances between the semantic information estimated from the corpus based on the distributional hypothesis and the semantic knowledge encoded in the ordinal ranking inequalities. In Rocktäschel et al. (2014), a similar approach was proposed to capture knowledge constraint as extra terms in the objective function for optimization.

In Figure 2, we show a diagram for the the overall SWE learning framework to incorporate semantic knowledge into the basic skip-gram word embeddings. Comparing with the previous work in (Xu et al., 2014) and (Faruqui et al., 2014), the proposed SWE framework is more general in terms of encoding the semantic knowledge for learning word embeddings. It is straightforward to show that the work in (Xu et al., 2014; Zweig, 2014; Faruqui et al., 2014) can be viewed as some special cases under our SWE learning framework.

3.3 Optimization algorithm for SWE

In this work, the proposed semantic word embeddings (SWE) are learned using the standard mini-batch stochastic gradient descent (SGD) algorithm. Furthermore, we adopt to use the cosine distance of the embedding vectors to compute the similarity between two words in the penalty term.

In the following, we show how to compute the derivatives of the penalty term for the SWE learning.

$$\begin{aligned}\frac{\partial \mathcal{D}}{\partial \mathbf{w}_t^{(1)}} &= \sum_{(i,j,k)\in S} \frac{\partial f\left(s_{ik} - s_{ij}\right)}{\partial \mathbf{w}_t^{(1)}} \\ &= \sum_{(i,j,k)\in S} f' \cdot \left(\delta_{ik}(t) \frac{\partial s_{ik}}{\partial \mathbf{w}_t^{(1)}} - \delta_{ij}(t) \frac{\partial s_{ij}}{\partial \mathbf{w}_t^{(1)}} \right)\end{aligned} \tag{9}$$

where $\delta_{ik}(t)$ and $\delta_{ij}(t)$ are computed as

$$\delta_{ik}(t) = \begin{cases} 1 & t = i \texttt{ or } t = k \\ 0 & \texttt{otherwise} \end{cases} \tag{10}$$

and for the hinge loss function $f(x)$, we have

$$f' = \begin{cases} 1 & (s_{ik} - s_{ij}) > \delta_0 \\ 0 & (s_{ik} - s_{ij}) \le \delta_0 \end{cases} \tag{11}$$

and the derivatives of the cosine similarity measure, $s_{ij} = \frac{\mathbf{w}_i^{(1)} \cdot \mathbf{w}_j^{(1)}}{|\mathbf{w}_i^{(1)}||\mathbf{w}_j^{(1)}|}$, with respect to a word vector, i.e., $\frac{\partial s_{ik}}{\partial \mathbf{w}_i^{(1)}}$, which can be derived as follows:

$$\frac{\partial s_{ij}}{\partial \mathbf{w}_i^{(1)}} = -\frac{s_{ij}\mathbf{w}_i^{(1)}}{|\mathbf{w}_i^{(1)}|^2} + \frac{\mathbf{w}_j^{(1)}}{|\mathbf{w}_i^{(1)}||\mathbf{w}_j^{(1)}|}. \quad (12)$$

The learning rate used for the SWE learning is the same as that for the skip-gram model. In each mini-batch of SGD, we sample terms in the same way as the skip-gram model. As for the constraints, we do not sample them but use all inequalities relevant to any words in a minibatch to update the model for the minibatch. Finally, by jointly optimizing the two terms in the combined objective function, we may learn a new set of word vectors encoding with ordinal semantic knowledge.

4 Experiments

In this section, we report all experiments conducted to evaluate the effectiveness of the proposed semantic word embeddings (SWE). Here we compare the performance of the proposed SWE model with the conventional skip-gram baseline model on four popular natural language processing tasks, including word similarity measure, sentence completion, name entity recognition, and synonym selection. In the following, we first describe the experimental setup, training corpora, semantic knowledge databases. Next, we report the experimental results on these four NLP tasks. Note that the SWE training codes and scripts are made publicly available at `http://home.ustc.edu.cn/~quanliu/`.

4.1 Experimental setup

4.1.1 Training corpora

In this work, we use the popular Wikipedia corpus as our training data to learn word embeddings for experiments on the word similarity task and the TOEFL synonym selection task. Particularly, we utilize two Wikipedia corpora with different sizes. The first corpus with a smaller size is a data set including the first one billion characters from Wikipedia[1], named as *Wiki-Small* in our experiments. The second corpus with a relatively large size is the latest Wikipedia dump[2], named as *Wiki-Large* in our experiments. Both Wikipedia corpora have been pre-processed by removing all the HTML meta-data and hyper-links and replacing the digit numbers with English words using the perl script from the Matt Mahoney's page[3]. After text normalization, the *Wiki-Small* corpus contains totally 130 million words, for which we create a lexicon of 225,909 distinct words appearing more than 5 times in the corpus. Similarly, the *Wiki-Large* corpus contains about 5 billion words, for which we create a lexicon of 228,069 words appearing more than 200 times.

For the other two tasks, sentence completion and name entity recognition, we use the same training corpora from the previous state-of-the-art work for fair comparisons. The training corpus for the sentence completion is the Holmes text (Zweig and Burges, 2011; Mikolov et al., 2013a). The training corpus for the name entity recognition task is the Reuters English newswire from RCV1 (Turian et al., 2010; Lewis et al., 2004). Refer to section 4.4 and section 4.5 for detailed descriptions respectively.

4.1.2 Semantic constraint collections

In this work, we use WordNet as the resource to collect ordinal semantic knowledge. WordNet is a large semantic lexicon database of English words (Miller, 1995), where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (usually called synsets). Each synset usually expresses a distinct semantic concept. All synsets in WordNet are interlinked by means of conceptual-semantic and/or lexical relations such as synonyms and antonyms, hypernyms and hyponyms.

In our experiments, we use the version WordNet-3.1 for creating the corresponding semantic constraints. In detail, we follow the following process to extract semantic similarity inequalities from WordNet and Thesaurus:

1. Based on the *Synonym Antonym Rule* described in section 2, for each word in vocabulary, find its synset and use the synonym and antonym relations to find all related synonymous and antonymous synsets. Note that the antonymous synset is selected as long as there exists an antonymous relation between any word in this synset and any word in an synonymous synset. After finding the synonymous and antonymous synsets

[1]http://mattmahoney.net/dc/enwik9.zip

[2]http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

[3]http://mattmahoney.net/dc/textdata.html

of the current word, the similarity inequalities could be generated according to the ranking rule. After processing all words, we have collected about 30,000 inequalities related to the synonym and antonym relations. Furthermore, we extract additional 320,000 inequalities from an old English dictionary (Fernald, 1896). In total, we have about 345,000 inequalities related to the synonym and antonym relations. This set of inequalities is denoted as *Synon-Anton* constraints in our experiments.

2. Based on the *Semantic Category Rule* and *Semantic Hierarchy Rule*, we extract another inequality set consisting of 75,000 inequalities from WordNet. We defined this collection as *Hyper-Hypon* constraints in our experiments.

In the following experiments, we just use all of these collected inequality constraints as is without further manually checking or cleaning-up. They may contain a very small percentage of errors or conflicts (due to multiple senses of a word).

4.1.3 Training parameter setting

Here we describe the control parameters used to learn the baseline skip-gram model and the proposed SWE model. In our experiments, we use the open-source *word2vec* toolkit[4] to train the baseline skip-gram model, where the context window size is set to be 5. The initial learning rate is set as 0.025 and the learning rate is decreased linearly during the SGD model training process. We use the popular negative sampling technique to speed up model training and set the negative sample number as 5.

To train the proposed SWE model, we use the same configuration as the skip-gram model to maximize $\mathcal{Q}'$. For the penalty term in eq. (7), we set $\delta_0 = 0$ for the hinge loss function. The semantic similarity between words is computed by the cosine distance. The combination coefficient β in eq. (8) is usually set to be a number between 0.001 and 0.3 in our experiments.

In the following four NLP tasks, the dimensionality of embedding vectors is different since we try to use the same settings from the state-of-the-art work for the comparison purpose. In the Word Similarity task and the TOEFL Synonym Selection task, we followed the state of the art work in (Xu et al., 2014), to set word embeddings to 300-dimension. Similarly, we refer to Bian et al. (2014) to set the dimensionality of word vectors to 600 for the Sentence Completion task. And we set the dimensionality of word vectors to 50 for the NER task according to (Turian et al., 2010; Pennington et al., 2014).

[4]https://code.google.com/p/word2vec.

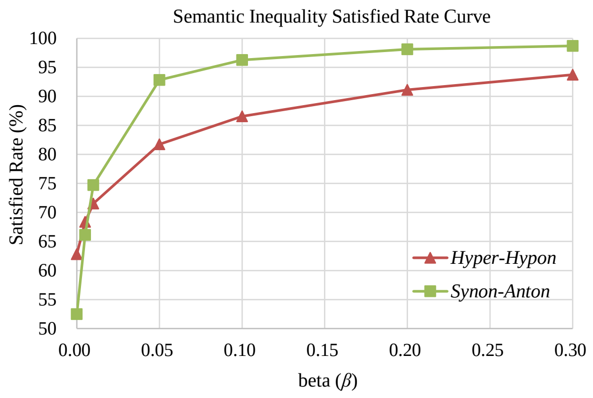

Figure 3: A curve of inequality satisfied rates (All models trained on the *Wiki-Small* corpus. *Hyper-Hypon* and *Synon-Anton* stand for different semantic constraint sets employed for training semantic word embeddings).

4.2 Semantic inequality satisfied rates

Here we first examine the inequality satisfied rates of various word embeddings. The inequality satisfied rate is defined as how many percentage of semantic inequalities are satisfied based on the underlying word embedding vectors. In Figure 3, we show a typical curve of the inequality satisfied rates as a function of β used in model training. This figure is plotted based on the *Wiki-Small* corpus. Two semantic constraint sets *Synon-Anton* and *Hyper-Hypon* created in section 4.1.2 are employed to learn semantic word embeddings.

In the framework of the proposed semantic word embedding method, we just need to tune one more parameter β, comparing with the skip-gram model. It shows that the baseline skip-gram ($\beta = 0$) can only satisfy about 50-60% of inequalities in the training set. As we choose a proper value for β, we may significantly improve the inequality satisfied rate, up to 85-95%. Although we can get higher inequality satisfying rate on the training set by increasing beta continuously, however, we do not suggest to use a big beta value because it would make the model overfitting. The major reason for this is that the constraints only cover a subset of words in vocabulary. Increasing the rate too much may screw up the entire word embeddings due to the sparsity of the constraints.

Meanwhile, we have found that the proposed SGD method is very efficient to handle a large number of inequalities in model training. When we use the total 345,000 inequalities, the SWE training is comparable with the baseline skip-gram model in terms of training speed. In the following, we continue to examine the SWE model on four popular natural language processing tasks, including word similarity, sentence completion, name entity recognition and the TOEFL synonym selection.

4.3 Task 1: Word Similarity Task

4.3.1 Task description

Measuring word similarity is a traditional NLP task (Rubenstein and Goodenough, 1965). Here we compare several word embedding models on a popular word similarity task, namely WordSim-353 (Finkelstein et al., 2001), which contains 353 English word pairs along with human-assigned similarity scores, which measure the relatedness of each word pair on a scale from 0 (totally unrelated words) to 10 (very much related or identical words). The final similarity score for each pair is the average across 13 to 16 human judges. When evaluating word embeddings on this task, we measure the performance by calculating the Spearman rank correlation between the human judgments and the similarity scores computed based on the learned word embeddings.

4.3.2 Experimental results

Here we compare the proposed SWE model with the baseline skip-gram model on the WordSim-353 task. Both word embedding models are trained using the Wikipedia corpora. We set the dimension of word embedding vectors to be 300. In Table 1, we have shown all the Spearman rank correlation results. The baseline results on this task include PPMI (Levy and Goldberg, 2014), GloVe (Pennington et al., 2014), and ESA-Wikipedia (Gabrilovich and Markovitch, 2007).

From the results in Table 1, we can see that the proposed SWE model can achieve consistent improvements over the baseline skip-gram model, no matter which training corpus is used. These results have demonstrated that, by incorporating semantic ordinal knowledge into the word vectors, the proposed semantic word embedding framework can capture much better semantics for many words. The SWE model using the *Wiki-Large* corpus has achieved the state-of-the-art performance on this task, significantly outperforming other popular word embedding methods, such as skip-gram and GloVe. Moreover, we also find that the *Synon-Anton* constraint set is more relevant than *Hyper-Hypon* for the word similarity task.

	Word Embeddings	**Result**
Others	SPPMI	0.6870
	GloVe (6 billion)	0.6580
	GloVe (42 billion)	0.7590
	ESA-Wikipedia	0.7500
Wiki-Small (0.13 billion)	Skip-gram	0.6326
	SWE + *Synon-Anton*	**0.6584**
	SWE + *Hyper-Hypon*	0.6407
	SWE + Both	0.6442
Wiki-Large (5 billion)	Skip-gram	0.7085
	SWE + *Synon-Anton*	**0.7274**
	SWE + *Hyper-Hypon*	0.7213
	SWE + Both	0.7236

Table 1: Spearman results on the WordSim-353 Task.

4.4 Task 2: Sentence Completion Task

4.4.1 Task description

The Microsoft sentence completion challenge has recently been introduced as a standard benchmark task for language modeling and other NLP techniques (Zweig and Burges, 2011). This task consists of 1040 sentences, each of which misses one word. The goal is to select a word that is the most coherent with the rest of the sentence, from a list of five candidates. Many NLP techniques have already been reported on this task, including N-gram model and LSA-based model proposed in (Zweig and Burges, 2011), log-bilinear model (Mnih and Teh, 2012), recurrent neural networks (RNN) (Mikolov, 2012), the skip-gram model (Mikolov et al., 2013a), a combination of the skip-gram and RNN model, and a knowledge enhanced word embedding model proposed by Bian et. al. (2014). The performance of all these techniques is listed in Table 2 for comparison.

In this work, we follow the the same procedure as in (Mikolov et al., 2013a) to examine the performance of our proposed semantic word embeddings (SWE) on this task. We first train 600-

dimension word embeddings based on a training corpus of 50M words provided by (Zweig and Burges, 2011), with and without using the collected ordinal knowledge. Then, for each sentence in the test set, we use the learned word embeddings to compute a sentence score for predicting all surrounding words based on each candidate word in the list. Finally, we use the computed sentence prediction scores to choose the most likely word from the given list to answer the question.

	System	Acc
Others	N-gram model	39.0
	LSA-based model	49.0
	Log-bilinear model	54.8
	RNN	55.4
	Skip-gram	48.0
	Skip-gram + RNN	58.9
Bian et al.	Skip-gram	41.2
	+ Syntactic knowledge	41.9
	+ Semantic knowledge	**45.2**
	+ Both knowledge	44.2
1 Iteration	Skip-gram	44.1
	SWE + *Synon-Anton*	47.9
	SWE + *Hyper-Hypon*	47.5
	SWE + Both	**48.3**
5 Iterations	Skip-gram	51.5
	SWE + *Synon-Anton*	55.7
	SWE + *Hyper-Hypon*	55.4
	SWE + Both	**56.2**

Table 2: Results on Sentence Completion Task.

4.4.2 Experimental results

In Table 2, we have shown the sentence completion accuracy on this task for various word embedding models. We can see that the proposed SWE model has achieved considerable improvements over the baseline skip-gram model. Once again, this suggests that the semantic knowledge represented by the ordinal inequalities can significantly improve the quality of the word embeddings. Besides, the SWE model significantly outperforms the recent work in (Bian et al., 2014), which considers syntactics and semantics of the sentence contexts.

4.5 Task 3: Name Entity Recognition

4.5.1 Task description

To further investigate the performance of semantic word embeddings, we have further conducted some experiments on the standard CoNLL03 name entity recognition (NER) task. The CoNLL03 NER dataset is drawn from the Reuters newswire. The training set contains 204K words (14K sentences, 946 documents), the test set contains 46K words (3.5K sentences, 231 documents), and the development set contains 51K words (3.3K sentences, 216 documents). We have listed the state-of-the-art performance in Table 3 for this task (Turian et al., 2010).

To make a fair comparison, we have used the exactly same experimental configurations as in (Turian et al., 2010), including the used training algorithm, the baseline discrete features and so on. Like the C&W model, we use the same training text resource to learn word vectors, which contains one year of Reuters English newswire from RCV1, from August 1996 to August 1997, having about 810,000 news stories (Lewis et al., 2004). Meanwhile, the dimension of word embeddings is set to 50 for all experiments on this task.

4.5.2 Experimental results

In our experiments, we compare the proposed SWE model with the baseline skip-gram model for name entity recognition, measured by the standard F1 scores. We present the final NER F1 scores on the CoNLL03 NER task in Table 3. The notation "Gaz" stands for gazetteers that are added into the NER system as an auxiliary feature. For the SWE model, we experiment two configurations by adding gazetteers or not (denoted by "IsGaz" and "NoGaz" respectively).

	System	Dev	Test	MUC7
Others	C&W	92.3	87.9	75.7
	C&W + Gaz	93.0	88.9	81.4
NoGaz	Skip-gram	92.6	88.3	76.7
	+ *Synon-Anton*	92.5	88.4	77.2
	+ *Hyper-Hypon*	92.6	**88.6**	**77.7**
	+ Both	92.6	88.4	77.5
IsGaz	Skip-gram	93.3	89.5	80.0
	+ *Synon-Anton*	93.1	89.6	80.7
	+ *Hyper-Hypon*	93.1	89.7	80.7
	+ Both	93.0	89.5	**80.8**

Table 3: F1 scores on the CoNLL03 NER task.

From the results shown in Table 3, we could find the proposed semantic word embedding (SWE) model can consistently achieve 0.8% (or more) absolute improvements on the MUC7 task no matter whether the gazetteers features are used or not. The proposed SWE model can also obtain 0.3% improvement in the CoNLL03 test set when no gazetteers is added into the NER system. However, no significant improvement is observed in this test set for the proposed SWE model after we add the gazetteers feature.

4.6 Task 4: TOEFL Synonym Selection

4.6.1 Task description

The goal of a synonym selection task is to select, from a list of candidate words, the semantically closest word for each given target word. The dataset we use for this task is the standard TOEFL dataset (Landauer and Dumais, 1997), which contains 80 questions. Each question consists of a target word along with 4 candidate lexical substitutes for selection.

The evaluation criterion on this task is the synonym selection accuracy which indicates how many synonyms are correctly selected for all 80 questions. Similar to the configurations on the word similarity task, all the experiments on this task are conducted on the English Wikipedia corpora. In our experiments, we set all the vector dimensions to 300.

4.6.2 Experimental Results

Corpus	Model	Accuracy (%)
Wiki-Small	Skip-gram	61.25
	+ *Synon-Anton*	**70.00**
	+ *Hyper-Hypon*	66.25
	+ Both	**71.25**
Wiki-Large	Skip-gram	83.75
	+ *Synon-Anton*	**87.50**
	+ *Hyper-Hypon*	85.00
	+ Both	**88.75**

Table 4: The TOEFL synonym selection task.

In Table 4, we have shown the experimental results for different word embedding models, learned from different Wikipedia corpora: *Wiki-Small* or *Wiki-Large*. We compare the proposed SWE with the baseline skip-gram model. From the experimental results in Table 4, we can see that the proposed SWE model can achieve consistent improvements over the baseline skip-gram model on the TOEFL synonym selection task, about 5-8% improvements on the selection accuracy. We find the similar performance differences between the SWE model trained with the *Synon-Anton* and *Hyper-Hypon* constraint set. The main reason would be that the synonym selection task is mainly related to lexical level similarity and less relevant to the hypernym-hyponym relations.

5 Conclusions and Future Work

Word embedding models with good semantic representations are quite invaluable to many natural language processing tasks. However, the current data-driven methods that learn word vectors from corpora based on the distributional hypothesis tend to suffer from some major limitations. In this paper, we propose a general and flexible framework to incorporate various types of semantic knowledge into the popular data-driven learning procedure for word embeddings. Our main contributions are to represent semantic knowledge as a number of ordinal similarity inequalities as well as to formulate the entire learning process as a constrained optimization problem. Meanwhile, the optimization problem could be solved by efficient stochastic gradient descend algorithm. Experimental results on four popular NLP tasks have all demonstrated that the propose semantic word embedding framework can significantly improve the quality of word representations.

As for the future work, we would incorporate more types of knowledge, such as knowledge graphs and FrameNet, into the learning process for more powerful word representations. We also expect that some common sense related semantic knowledge may be generated as ordinal inequality constraints by human annotators for learning semantic word embeddings. At the end, we plan to apply the SWE word embedding models for more natural language processing tasks.

Acknowledgments

This work was supported in part by the Science and Technology Development of Anhui Province, China (Grants No. 2014z02006) and the Fundamental Research Funds for the Central Universities. Special thanks to Zhigang Chen, Ruiji Fu and the anonymous reviewers for their insightful comments as well as suggestions. The authors also want to thank Profs. Wu Guo and Li-Rong Dai for their wonderful help and supports during the experiments.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Knowledge-powered deep learning for word embedding. In *Machine Learning and Knowledge Discovery in Databases*, pages 132–148. Springer.

Zhigang Chen, Wei Lin, Qian Chen, Xiaoping Chen, Si Wei, Xiaodan Zhu, and Hui Jiang. 2015. Revisiting word embedding for contrasting meaning. In *Proceedings of ACL*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of ACL*, pages 1370–1380.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. In *Proceedings of the NIPS Deep learning and representation learning workshop*.

James Champlin Fernald. 1896. *English synonyms and antonyms*. Funk & Wagnalls Company.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of WWW*, pages 406–414. ACM.

Ronald A Fisher. 1936. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of IJCAI*, volume 7, pages 1606–1611.

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

Geoffrey E Hinton, James L McClelland, and David E Rumelhart. 1986. Distributed representations. In *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations*, pages 77–109. MIT Press.

Dan Jurafsky and James H Martin. 2000. *Speech & language processing*. Pearson Education India.

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*, 104(2):211.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceedings of NIPS*, pages 2177–2185.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Tomáš Mikolov. 2012. *Statistical language models based on neural networks*. Ph.D. thesis, Ph. D. thesis, Brno University of Technology.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

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 ICML*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. *Proceedings of EMNLP*, 12:1532–1543.

Tim Rocktäschel, Matko Bošnjak, Sameer Singh, and Sebastian Riedel. 2014. Low-dimensional embeddings of logic. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 45–49, Baltimore, MD, June. Association for Computational Linguistics.

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

Holger Schwenk. 2007. Continuous space language models. *Computer Speech & Language*, 21(3):492–518.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, pages 3104–3112.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394. Association for Computational Linguistics.

Peter D Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of CIKM*, pages 1219–1228. ACM.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of ACL*, volume 2, pages 545–550.

Geoffrey Zweig and Christopher JC Burges. 2011. The microsoft research sentence completion challenge. Technical report, Technical Report MSR-TR-2011-129, Microsoft.

Geoffrey Zweig. 2014. Explicit representation of antonymy in language modelling. Technical report, Technical Report MSR-TR-2014-52, Microsoft.