Semantic Structure-based Word Embedding by Incorporating Concept Convergence and Word Divergence

Qian Liu[1,2], **Heyan Huang**[1,3*], **Guangquan Zhang**[2], **Yang Gao**[1,4], **Junyu Xuan**[2], **Jie Lu**[2]

1. Department of Computer Science, Beijing Institute of Technology, China
2. DeSI Lab, Centre for Artificial Intelligence, University of Technology Sydney, Australia
3. Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications, China
4. Beijing Advanced Innovation Center for Imaging Technology, Captial Normal University, China

Abstract

Representing the semantics of words is a fundamental task in text processing. Several research studies have shown that text and knowledge bases (KBs) are complementary sources for word embedding learning. Most existing methods only consider relationships within word-pairs in the usage of KBs. We argue that the structural information of well-organized words within the KBs is able to convey more effective and stable knowledge in capturing semantics of words. In this paper, we propose a semantic structure-based word embedding method, and introduce concept convergence and word divergence to reveal semantic structures in the word embedding learning process. To assess the effectiveness of our method, we use WordNet for training and conduct extensive experiments on word similarity, word analogy, text classification and query expansion. The experimental results show that our method outperforms state-of-the-art methods, including the methods trained solely on the corpus, and others trained on the corpus and the KBs.

Introduction

Understanding and representing the sense of text is a fundamental task in both information retrieval (IR) and natural language processing (NLP). Previous research has expended great effort on constructing distributed representations of words (also known as word embedding) as the atomic components of text by embedding the semantic and syntactic properties of the surface text into low-dimensional dense vectors. Trained word embeddings have achieved overwhelming success in various real-world applications, e.g., document retrieval (Bengio, Courville, and Vincent 2013; Passalis and Tefas 2016; Roy et al. 2016), text classification (Lampos, Zou, and Cox 2017), question answering (Shen et al. 2017), and sentiment classification (Bollegala, Mu, and Goulermas 2016).

Most of the research directs attention entirely towards learning word representation methods from a large unlabeled corpus, such as *prediction-based* methods (Collobert et al. 2011; Mikolov et al. 2013b; 2013a; Cao and Lu 2017) which learn word representation by predicting the co-occurrence of words in the given context, and

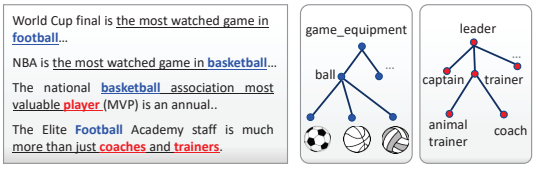

Figure 1: We mark two types of words in the sentences with blue color and red color, respectively. The underlined words are their context in the corpus. The graphs in the right are their semantic structures generated from WordNet.

counting-based methods (Pennington, Socher, and Manning 2014) which learn word representations through global matrix factorization based on a count of co-occurring words. These corpus-based methods mainly consider a word's co-occurrence information and, therefore, generally learn similar embeddings for words with similar contexts.

In the past few years, some efforts have focused on learning word representation beyond the corpus, and considered external knowledge bases constructed by human experts, such as semantic lexicons and concept graphs (Ponzetto and Navigli 2010; Bollegala, Maehara, and Kawarabayashi 2015; Liu et al. 2015a; Bollegala et al. 2016; Goikoetxea, Agirre, and Soroa 2016). Most previously proposed methods simply use relations within word-pairs, e.g., constraining words belonging to one semantic category (Yu and Dredze 2014), or constructing a regularizer to model words in particular semantic relations (Bollegala et al. 2016). As such, this work did not fully explore the comprehensive structures in the KBs.

In this paper, we argue that effective word embeddings should contain the semantic structures within the knowledge base. We illustrate how the semantic structures can be a complementary source for word embeddings in Fig.1. As shown in the sentences, *football*, *basketball*, *trainer*, *coach* usually share similar context, and tend to have similar representations in the corpus-based methods. While the semantic structures in the right side clearly define these words with different semantic granularities and abstractions, i.e., these four words are located in two different subgraphs, showing that they belong to different concepts; *football* and *basketball* are not directly linked in the subgraph, showing that

*Corresponding author

they hold different attributes. On the other hand, compared with relations in word-pairs, comprehensively modeling a word's structural features with its directly linked and indirectly linked words in the KBs could be more stable and reliable(Xuan et al. 2016; 2017).

To this end, we propose a **se**ma**n**tic **s**tructure-based word **e**mbedding method called SENSE. Moreover, we introduce *concept convergence* and *word divergence* to implement semantic structure modeling in the word embedding learning process. The basic idea can be intuitively explained as *football* and *basketball* are related to *ball* (concept convergence), but they also hold different attributes since they are indirectly linked in the graph (word divergence). We evaluate our word embedding method using extensive intrinsic and extrinsic evaluations. The experimental results show that modeling semantic structures in the knowledge base by incorporating concept convergence and word divergence makes embeddings significantly more powerful, and results in consistent performance improvement across real-world applications.

This paper departs from previous work in that it explores global structural information of words in the usage of knowledge base, not the local relations that exist between two words. The main contributions can be summarized as follows:

- We design a novel approach for learning word embedding that considers relatively stable and reliable semantic structures within the KBs.
- We design the principle of preserving semantic structures by converging words to their concept on the upper level and diverging words on the same sense level. We show that this principle is effective and easy to implement into the word embedding training process.
- To validate this method, we conducted extensive experiments on semantic property testing, document retrieval, and text classification. The experiment results show that the proposed method significantly outperforms the state-of-the-art methods.

Related Work

Word representation aims to learn a transformation of each word from raw text data to a representation that is mathematically and computationally convenient to process in text processing tasks. The last few years have seen the development of distributed word representation learning methods purely based on the co-occurrence information in a corpus (Bengio et al. 2003; Mnih and Hinton 2008; Collobert et al. 2011; Mikolov et al. 2013b; 2013a; Mnih and Kavukcuoglu 2013; Lebret and Collobert 2014; Pennington, Socher, and Manning 2014; Barkan 2017; Cao and Lu 2017). Some recent studies throw light on the semantic knowledge stored in the KBs, showing that the KBs can potentially assist the word embedding learning process.

Several studies use combined methods to fit pre-trained word embeddings with the given external resource, making no assumptions about how the input embeddings were constructed. For example, the *Retrofit* method (Faruqui et al. 2015) refines word representations using relational information from semantic lexicons. The method encourages linked words to have similar vector representations which are then embedded in a semantic network that consists of linked word senses in a continuous-vector word space. Johansson et al.(2015) presented a method to embed a semantic network into a pre-trained word embedding, considering that vectors for polysemous words can be decomposed into a convex combination of sense vectors and the vector for a sense is kept similar to those of its neighbors in the network. Goikoetxea et al.(2016) learned word representations from text and WordNet independently, and then explored both simple and sophisticated methods to combine them, showing that a simple concatenation of independently learned embeddings outperforms more complex combination techniques in word similarity and relatedness datasets.

In contrast to the combined methods, several studies have jointly leveraged semantic lexicons and corpus-based methods. The RCM method (Yu and Dredze 2014) is a relation constrained model which introduces a training objective that incorporates both a neural language model objective and a semantic knowledge objective. In the RCM method, the knowledge base functions as word similarity information to improve the performance of word embedding. Xu et al.(2014) leveraged both relational and categorical knowledge to produce word representation (RC-NET), combining this with the Skip-gram method. Liu et al.(2015a) represents semantic knowledge as a number of ordinal similarity inequalities of related word pairs to learn semantic word embedding (SWE). Bollegala et al.(2015) proposed a method that considers semantic relations in which they co-occur to learn word representations. And they (Bollegala et al. 2016) also proposed a joint word representation learning method that simultaneously predicts the co-occurrences of two words in a sentence, subject to the relational constraints given by a semantic lexicon . Although these studies consider the semantic information from an external knowledge base in the learning process, they do not leverage high-quality semantic structures to improve word embeddings.

Our work in this paper can be categorized as a joint learning method that incorporates both co-occurrence information and semantic structures. In contrast to the aforementioned research, we leverage the semantic structure information in the KBs. In our method, we construct multi-level structures from the knowledge base to express semantic granularity and abstraction. Moreover, we design principles of concept convergence and word divergence to implement semantic structures into the word embedding learning process.

Semantic Structure-based Word Embedding

The Basic Idea

Given a corpus $\mathcal{C}$ and a knowledge base $\mathcal{G}$ as input, the SENSE method learns a d dimensional vector $\mathbf{x}_w \in \mathbb{R}^d$ for each word w in the corpus. Any KB that captures the relationships between words in a hierarchically-organized manner could be used to generate semantic structures, such as WordNet (Miller 1995; Fellbaum 1998), Freebase (Bol-

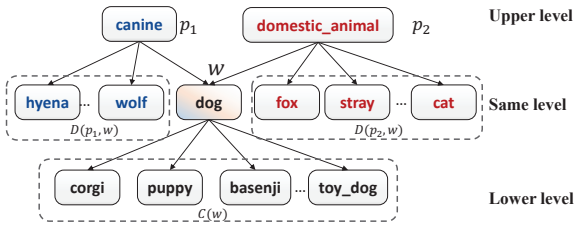

Figure 2: An example of the three-level semantic structures of the word *dog* in WordNet.

lacker et al. 2008), and PPDB (Ganitkevitch, Durme, and Callison-Burch 2013). In this paper, we use WordNet to describe the method and conduct the experiments.

The KB is defined as a directed graph $\mathcal{G} = (V, E)$, where the set of vertices V denotes words, and the set of edges E denotes the semantic relations between the pairs of vertices. Intuitively, a vertex's structure information in a directed graph can be covered by exploiting its parent vertices, brother vertices, and child vertices. Fig. 2 visualizes the structures of the word *dog* in WordNet. Our ideas for modeling the structures were inspired by the observations in the nature language.

First, words directly linked in semantic structures share the same attributes. For example, *canine* is the parent of *dog* and *wolf*, and *canine* can be regarded as a concept that represents the common attributes of a *dog* and a *wolf*. These directly linked words tend to converge, and the child words tend to be close to the parent word. Thus, we assume that:

Assumption 1 *Concept convergence: The upper level is regarded as the concept of its lower level. The center of all words in the lower level tends to converge to their upper-level word.*

Second, brother words in semantic structures are indirectly linked and are located in the same level. They tend to be diverged, giving the areas of different words a distinct positioning for different attributes. For example, *wolf* and *dog* are close to *canine* as they share the same attribute, but they should be separated from each other since they also hold significantly different attributes. Thus, we assume that:

Assumption 2 *Word divergence: Words in the same level hold distinctive attributes, and they tend to be diverged.*

The Proposed Method

A variety of corpus-based methods have been proposed to learn word representations by optimizing the prediction ability between words and contexts. We follow the Word2Vec method, which uses extremely computationally efficient log-linear models to produce high-quality word embeddings. The Word2Vec method applies a sliding window moving on the corpus, and the central word is the target word and the others are context words. There are two models: the CBOW model uses the average/sum of context words as input to predict the target; the Skip-gram model uses the target word as input to predict each context word. To simplify, we represent the objective of each prediction as

$$\mathcal{L}_{context} = Pr(w|\mathbf{c}) = \frac{\exp(\mathbf{x}_w \cdot \mathbf{c})}{\sum_{w' \in \mathcal{V}} \exp(\mathbf{x}_{w'} \cdot \mathbf{c})}, \quad (1)$$

where w is the predicted word, $\mathbf{c}$ is the vector of input word/words, $\mathbf{x}_{w'} \in \mathbb{R}^d$ is the vector representation of the word w' in the vocabulary $\mathcal{V}$.

The objective of the SENSE method is to train word representations that are not only good at predicting its context words, but are also good at modeling concept convergence and word divergence. Let w represent the predicted word in each prediction task. We detail how to represent structural information of word w in $\mathcal{G}$.

Specially, we define $\mathcal{G}$ using WordNet, where words are grouped into sets of cognitive synonyms (denoted as synsets), and synsets are interlinked by hyponym-hypernym relations (i.e., general terms and specific kinds). We observe that WordNet is a complex hierarchical graph of synsets: (1) each word points to at least one synset. Hence, there is a many-to-many relationship between synsets and words; (2) the synset would have more than one parent in WordNet. In our method, we model the semantic structures on the granularity of synsets. Formally, given a word, we denote its synset collection as $S = \{w^1, \ldots, w^k\}$, where $w^i (1 \leq i \leq k)$ represents one synset of the word, denoted as w for brevity. Then for each synset of word w, we exploit the following three-level features that capture varying granularity semantic structures:

- Let $P(w) = \{p_1, \ldots, p_{|P|}\}$ represent the collection of words on the upper level of word w, where $p_i \in V$, and the edge $< p_i, w >$ exists in E.
- Words on the same level of w are divided into $|P(w)|$ subsets regarding different parent words. Each subset is denoted as $D(p_i, w) = \{u_1, \ldots, u_{|D|}\}$, where $u \in V$, and the edge $< p_i, u >$ exists in E.
- Words on the lower level are specific terms of w, denoted as $C(w) = \{v_1, \ldots v_{|C|}\}$, where $v \in V$, and the edge $< w, v >$ exists in E.

Based on the concept convergence assumption described above, we assume that w should be close to the center of words on the lower level of w (i.e., words in $C(w)$). The training objective is defined to maximize the following function:

$$\mathcal{L}_c = \sum_{S(w)} cos(\mathbf{x}_w, \frac{1}{|C|} \sum_{v \in C(w)} \mathbf{x}_v), \quad (2)$$

where $|C|$ is the size of collection $C(w)$. Here $cos(\cdot, \cdot)$ represents the similarity measure function. Following the recommendations in prior work on word similarity measurement, we apply the cosine similarity of a pair of words w_a, w_b by computing

$$cos(\mathbf{x}_{w_a}, \mathbf{x}_{w_b}) = \frac{\mathbf{x}_{w_a}^T \cdot \mathbf{x}_{w_b}}{|\mathbf{x}_{w_a}| \cdot |\mathbf{x}_{w_b}|}. \quad (3)$$

The word divergence assumption is defined as enlarging the distance between w and words in the same level with

w (i.e., words in $D(\cdot, w)$), and the training objective is to minimize the following function:

$$\mathcal{L}_d = \sum_{S(w)} \sum_{p_i \in P(w)} \sum_{u \in D(p_i, w)} cos(\mathbf{x}_w, \mathbf{x}_u), \quad (4)$$

where $P(w)$ is the collection of w's upper level. Because some words have many brother words in KBs, we randomly select several words in the training step. We find that selecting five words is an acceptable trade-off between the method's performance and training speed.

As mentioned before, we integrate the context information and the semantic structure information into a unified framework. Then the new optimization objective is

$$\mathcal{L} = \max_{\Theta}(\mathcal{L}_{context} + \alpha\mathcal{L}_c - \beta\mathcal{L}_d), \quad (5)$$

where Θ is a set of all the parameters related to this task, α and β are hyper-parameters, which control the contributions of semantic structures in word embedding learning.

Using the optimization method in (Mikolov et al. 2013b), we apply negative sampling to solve the context prediction function. If the predicted word w has semantic structures in the KB, the corresponding optimization process for modeling the semantic structures will be activated. The optimization is as follows:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{x}_w} &= \alpha \frac{\partial \mathcal{L}_c}{\partial \mathbf{x}_w} - \beta \frac{\partial \mathcal{L}_d}{\partial \mathbf{x}_w} = \sum_{S(w)} (\alpha \frac{\partial cos(\mathbf{x}_w, \overline{\mathbf{x}})}{\partial \mathbf{x}_w} \\
&\quad - \beta \sum_{p_i \in P(w)} \sum_{u \in D_{(p_i, w)}} \frac{\partial cos(\mathbf{x}_w, \mathbf{x}_u)}{\partial \mathbf{x}_w}), \\
\frac{\partial \mathcal{L}}{\partial \mathbf{x}_v} &= \alpha \frac{\partial \mathcal{L}_c}{\partial \mathbf{x}_v} = \sum_{S(w)} \alpha \frac{\partial cos(\mathbf{x}_w, \overline{\mathbf{x}})}{\partial \overline{\mathbf{x}}}, \quad (6) \\
\frac{\partial \mathcal{L}}{\partial \mathbf{x}_u} &= -\beta \frac{\partial \mathcal{L}_d}{\partial \mathbf{x}_u} \\
&= \sum_{S(w)} \sum_{p_i \in P(w)} \sum_{u \in D(p_i, w)} -\beta \frac{\partial cos(\mathbf{x}_w, \mathbf{x}_u)}{\partial \mathbf{x}_u},
\end{aligned}$$

where w is the predicted word, u is the word in $D(\cdot, w)$, v is the word in $C(w)$, and $\overline{\mathbf{x}}$ is the average vector of words in $C(w)$. Since we apply the cosine distance to compute the similarity between two words, the optimization can be derived as follows:

$$\frac{\partial cos(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} = -\frac{S_{ij} \cdot \mathbf{x}_i}{|\mathbf{x}_i|^2} + \frac{\mathbf{x}_j}{|\mathbf{x}_i| \cdot |\mathbf{x}_j|}, \quad (7)$$

where $S_{i,j} = \frac{\mathbf{x}_i^T \cdot \mathbf{x}_j}{|\mathbf{x}_i| \cdot |\mathbf{x}_j|}$.

In our implementation, the optimization process is conducted through SGD in a mini-batch mode, with a computational complexity comparable to the optimization process in the Word2Vec method. The pseudo code for our word embedding learning method is shown in Algorithm.1.

Experiments and Results

In this section, we first evaluate the SENSE method's ability to capture semantic and syntactic properties of words. Then, we conduct experiments on the text classification task and the query expansion task, showing that the proposed method boosts performance in real-world applications. The source code of our method is available in the GitHub[1].

Algorithm 1 SENSE method.

Require: WordNet G, Corpus C, dimensionality d of the word embeddings, word vocabulary $\mathcal{V}$
Ensure: Embeddings $\mathbf{x}_w \in \mathcal{R}^d$ of all words in the vocabulary $\mathcal{V}$.
1: **Initialization:** randomly set $\mathbf{x}_w \in \mathcal{R}^d$ for all words $w \in \mathcal{V}$; generate the semantic structures of each word in G; constructing T prediction tasks using a sliding window.
2: **for** t = 1, 2, . . . , T **do**
3: optimizing $\mathcal{L}_{context}$ using negative sample method introduced in (Mikolov et al. 2013b)
4: **if** w in G **then**
5: use Eq.(6) to update $\mathbf{x}_w$, $\mathbf{x}_u$, $\mathbf{x}_v$.
6: **end if**
7: **end for**
8: **return** $\mathbf{x}_w$ for all words $w \in \mathcal{V}$.

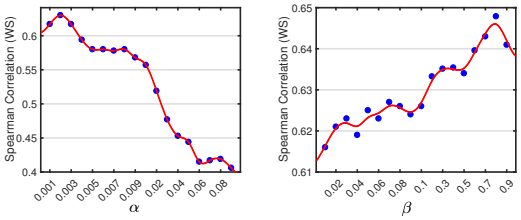

Figure 3: Performance of the SENSE method with varying parameters of α and β.

Initialization and Parameters

We utilize WordNet (version 3.0) as the KB and use the semantic structure information when words are linked using hypernym-hyponym relation. Since only nouns and verbs hold a hypernym-hyponym relation in WordNet, we extract all the nouns and verbs in WordNet to construct the graph $\mathcal{G}$, resulting in 66,765 nouns with 82,115 synsets and 7,440 verbs with 13,767 synsets.

There are two hyper-parameters in the SENSE method, i.e. α and β in Eq.(5), which control the contributions of the semantic structures to the joint learning process. We carefully tune these parameters by fixing one and varying the other. The parameters corresponding to the best word similarity metric value (detailed in next subsection) are used to report the final settings. As shown in Fig.3, the SENSE method reaches optimal performance when α = 0.002 and β = 0.8. We follow the optimal settings in this work, with recommended settings of $\alpha \in (0.001, 0.003)$ and $\beta \in (0.7, 0.9)$.

For a fair comparison, all word embeddings adhere to the

[1]https://github.com/qianliu0708/SENSE

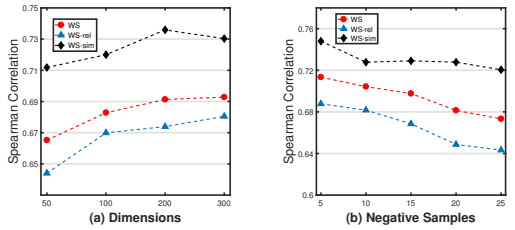

Figure 4: Performance over varying parameters on the WordSim 353 dataset.

following settings: the dimensionality of vectors is 300, the size of the context window is 5, the number of negative samples is 5, and all KB-enhanced methods are trained using WordNet. Specially, to understand the robustness of our method, we explore the relation between the performance of our method on the word similarity task with varying number of dimensions and negative samples. As shown in Fig.4, we observe that our method is stable when the dimension is set to a value between 100 and 300. The best performance can be obtained when the dimension is set to 300. Regarding the size of negative samples, our method obtains optimal results when the number of negative samples is set to 5, and the performance of our method degrades when the number of negative samples is too large.

Word Similarity and Word Analogy

Baselines We compare the SENSE method against two classes of baselines:
(1) **The corpus-based methods** which train word embeddings solely on the corpus. We use the current state-of-the-art methods, including:

- CBOW[2](Mikolov et al. 2013b) is a neural network language model which learns word embeddings by maximizing the conditional probability of a target word given the context.
- Skip-gram[3](Mikolov et al. 2013b) is a neural network language model which learns word embeddings by maximizing the conditional probability of a context word given the target word.
- GloVe[4] (Pennington, Socher, and Manning 2014) is a state-of-the-art matrix factorization method. It leverages global count information aggregated from the entire corpus as word-word occurrence matrix to learn word embeddings.

(2) **The KB-enhanced methods** which train word embeddings both on the corpus and the KBs. To make a comprehensive comparison, we compare the SENSE method against popular and powerful methods which also use the external KBs, including:

- RCM[5](Yu and Dredze 2014) is a relational constrained word embedding method. It incorporates both the objective of context prediction (following CBOW method and Skip-gram method) and the objective which constrained the relations from the KBs.
- Retrofit[6] (Faruqui et al. 2015) is a popular method that refines pre-trained word embeddings using relational information from the KBs.
- Jointreps[7] (Bollegala et al. 2016) is a method jointly trained on a word co-occurrence matrix from the corpus (following the GloVe method) and semantic relations from KBs.

Datesets and Settings We intrinsically evaluate our method on two standard tasks: the word similarity task by predicting the semantic similarity between words, and the word analogy task by predicting proportional analogies consisting of two pairs of words. The training corpus for all methods is a subset of the Wikipedia corpus, which contains 16 million words and 71,291 distinct words.

We conduct the word similarity task using the following benchmark datasets: **MC** (30 word-pairs) (Miller and Charles 1991), **MEN** (3000 word-pairs) (Bruni et al. 2012), **RG** (65 word-pairs) (Luong, Socher, and Manning 2013), **VERB** (143 word-pairs) (Baker, Reichart, and Korhonen 2014), **WS** (353 word-pairs), and its similarity subset (**WS-sim**) and relatedness subset (**WS-rel**) (Agirre et al. 2009). Each word-pair in these benchmark datasets has a human-assigned similarity score. We calculate cosine similarity between the vectors of two words forming a test item, and report Spearmans rank correlation coefficient (Spearman 1904) between the rankings produced by the word embedding methods against the human rankings.

To assess the methods ability to perform semantic deduction, we evaluate word embedding methods using a word analogy task introduced by Mikolove (2013b). The task defines a comprehensive test that contains 19,544 questions divided into a semantic subset and a syntactic subset. The semantic subset contains five types of analogy questions about people or places, such as *"America is to New York as Australia is to ?"*. The syntactic subset contains nine types of analogy questions regarding verb tenses or forms of adjectives, such as *"good is to better as bad is to ?"*.

For each question, given w_1, w_2, w_3, it requires a fourth word w_4 to be generated to satisfy the question *"w_1 is to w_2 that is similar to w_3 is to w_4"*. The method we use to answer the question is by finding the optimal word using the following function:

$$v^* = \underset{v}{\operatorname{argmax}}\ cos(v, v_2) - cos(v, v_1) + cos(v, v_3), \quad (8)$$

where v_1, v_2, and v_3 are the embeddings of word w_1, w_2, w_3, and $cos(\cdot, \cdot)$ is the cosine similarity function. The best embedding of v^* is regarded as the answer.

[2]http://code.google.com/p/word2vec
[3]http://code.google.com/p/word2vec
[4]http://nlp.stanford.edu/projects/glove/
[5]https://github.com/Gorov/JointRCM
[6]https://github.com/mfaruqui/retrofitting
[7]https://github.com/Bollegala/jointreps

Methods	Word Similarity							Word Analogy		
	MC	MEN	RG	VERB	WS	WS-rel	WS-sim	Sem	Syn	Tot
GloVe	0.459	0.506	0.374	0.293	0.509	0.546	0.538	63.5	33.3	56.8
Retrofit-GloVe	0.566	0.526	0.469	0.225	0.539	0.517	0.599	45.3	24.1	42.2
Jointreps	0.394	0.429	0.340	0.308	0.465	0.384	0.534	11.5	6.9	8.8
CBOW	0.641	0.658	0.654	0.402	0.638	0.615	0.708	48.2	41.6	48.7
RCM-CBOW	0.492	0.411	0.448	0.247	0.496	0.399	0.569	21.9	11.5	15.1
Retrofit-CBOW	0.677	0.654	0.673	0.365	0.639	0.612	0.711	36.5	38.5	39.1
SENSE-CBOW	**0.692**	**0.665**	**0.685**	0.402	**0.688**	**0.657**	**0.719**	**49.9**	**42.2**	**49.9**
Skip-gram	0.640	0.676	0.682	0.343	0.631	0.621	0.695	62.4	33.6	56.0
RCM-Skipgram	0.478	0.416	0.418	0.261	0.481	0.393	0.544	21.8	10.9	14.7
Retrofit-Skipgram	0.599	0.576	0.622	0.134	0.569	0.467	0.637	34.9	25.4	35.6
SENSE-Skipgram	**0.678**	**0.678**	**0.686**	**0.374**	**0.694**	**0.674**	**0.733**	**63.9**	**33.8**	**57.2**

Table 1: Results on the word similarity task and the word analogy task. The word embedding methods are divided into three groups. Bold scores are the best within the groups. Underlined scores are the best overall.

Results Table 1 shows the evaluation results for both the word similarity task and the word analogy task. From the results, we observe that:

(1) We observe that most KB-enhanced methods perform better compared to their baseline methods (e.g., Retrofit-CBOW v.s. CBOW), while the RCM method and the Jointreps method do not perform better than their corresponding baseline methods. This observation demonstrates that external KBs can boost the performance of word embeddings, but the methods of how to extract and model the semantic information may directly affect the performances. Our SENSE method significantly outperforms over all the baseline methods, which means that modeling semantic structures by concept convergence and word divergence is reasonable and effective.

(2) The SENSE method reports the best results in seven word similarity datasets and the word analogy dataset. In particular, the improvements reported by the SENSE method are statistically significant on MC, RG, WS, WS-rel, and WS-sim. We attribute the success of our method to its power in modeling structural information in the word embedding learning process.

(3) For the task of word analogy, the GloVe method is a much stronger baseline than the others. It is fair to say that the global counting information is more accurate for semantic deduction compared to local co-occurrence information. The SENSE-Skipgram model still performs better than the GloVe method, demonstrating the generality and effectiveness of our method. It also implies that semantic structures are more reliable and stable knowledge than the relationship between word-pairs, and structural information can capture a word's latent relation in a global view.

Text Classification

We investigate the effectiveness of the SENSE method for text classification. The experiment is conducted on the 20NewsGroup[8] dataset. We use the bydate version which contains 18,846 documents from 20 different newsgroups.

Methods	Acc.	Prec.	Rec.	F1
LDA	72.2	70.8	70.7	70.0
BOW	79.7	79.5	79.0	79.0
PV-DM	72.4	72.1	71.5	71.5
PV-DBOW	75.4	74.9	74.3	74.3
TWE	71.7	70.9	70.4	69.7
GloVe	62.3	61.2	61.1	60.5
CBOW	78.1	77.4	77.1	77.0
Skip-gram	80.2	79.6	79.1	79.0
Retrofit-CBOW	75.6	75.9	73.5	72.1
Retrofit-Skipgram	77.4	77.9	75.5	74.3
SENSE-CBOW	**81.4**	**80.8**	**80.3**	**80.2**
SENSE-Skipgram	**81.7**	**81.2**	**80.6**	**80.6**

Table 2: Evaluation results of multi-class text classification. Bold scores denote the SENSE method outperforms the corresponding baseline methods. Underlined scores are the best overall.

The dataset is separated into a training set of 11,314 documents and a test set of 7,532 documents. All documents are joined together as a corpus for training word embeddings. We tokenize the corpus with the Stanford Tokenizer[9] and convert it to lower case, then removed the stop words. The corpus is 30.4M and contains 6.3 million words.

We consider the following baselines, BOW, LDA, TWE (Liu et al. 2015b), GloVe, Word2Vec, Retrofit and PV (Le and Mikolov 2014). The BOW method represents each document as a bag of words and the weighting scheme is TFIDF (the top 50,000 words are selected). The LDA represents each document as its inferred topic distribution. We set the number of topics as 80. The PV method is an unsupervised learning algorithm that learns vector representations for documents by predicting words in the document, including distributed memory model (PV-DM) and the distributed bag-of-words model (PV-DBOW). For word embedding methods, we construct document embeddings $\mathbf{d}$ by simply averaging all word embeddings in the given document, i.e.,

[8]http://qwone.com/ jason/20Newsgroups/.

[9]https://nlp.stanford.edu/software/tokenizer.shtml

$\mathbf{d} = \sum_{w \in d} \mathbf{x}_w$, where w is a word in document d, and $\mathbf{x}_w$ is the word embedding of word w. We regard document embedding vectors as a document feature and train a linear classifier using Liblinear[10] (Fan et al. 2008), since the feature size ($d = 300$) is large, and the Liblinear can quickly train the linear classifier with high dimension features. The classifier is then used to predict the class labels of documents in the testing set. We report the macro-averaging accuracy, precision, recall, and F1−measure for comparison.

Table 2 shows the evaluation results of text classification. We observe that the SENSE-Skipgram method significantly outperforms all baseline methods, showing that our method better captures the semantic information of documents. Both SENSE-CBOW and SENSE-Skipgram outperform their basic methods, especially SENSE-CBOW achieves a 3.3% improvement over the CBOW method. Whereas two Retrofit methods do not perform as well as the basic Word2Vec method. This observation shows the superiority and generality of our SENSE method with modeling semantic structures in the word embedding learning process.

Query Expansion

We evaluate the performance of the SENSE method in query expansion for the information retrieval task. The experiment is conducted on the Reuters Corpus Volume 1 (RCV1) dataset, which contains 806,791 documents. We combine the *title* and *text* parts of all documents to construct a training corpus, and then tokenized the training corpus with the help of the Stanford Tokenizer tool and convert every word to lower case. The corpus totals 16 million words.

The documents are divided into 50 collections, and each collection contains a training set and a test set. We implement the query expansion as follows: (1) we generated original queries by selecting the top 10 words in each collection, using the weighting scheme BM25; (2) then for each query q, we use word embeddings to select the top 5 most similar words with cosine similarity as its expansion words; (3) each expansion word w is associated with a weight as $w(q) * cos(\mathbf{q}, \mathbf{w})$, where $w(q)$ is the weight (BM25 score) of the original query, and $cos(\mathbf{q}, \mathbf{w})$ is the cosine similarity of the embeddings of the query and the expansion word. Finally, we construct an expanded query set Q^* which contains original queries and expanded words. Each query q in Q^* is associated with a weight, denoted as $w(q)$.

We retrieve the documents using the set Q^*. For each document d, its relevance score s to the query set is computed as $s = \sum_{q \in Q^*} f(q) * w(q)$, if $q \in d$, $f(q) = 1$; otherwise $f(q) = 0$. We report four standard evaluation metrics: the average precision of the top 10 documents (P@10) and top 20 documents (P@20), the mean average precision (MAP), and the F1−measure.

Table 3 reports the results achieved by the proposed method and the baselines. We observe that all the query expansion methods significantly outperform the BM25 method, which indicates the effectiveness of employing word embeddings for query expansion. According to the table, the SENSE-CBOW method consistently outperforms all compared methods, and our methods significantly outperform their corresponding baseline methods. While other KB-enhanced method, i.e. Jointreps and Retrofit, perform slightly better than their baseline methods. Moreover, compared to the GloVe method and the TWE method, the SENSE method achieves remarkable improvements. This observation also indicates that semantic structures are more effective in capturing semantic features than collecting topical information and global co-occurrence information.

[10]https://www.csie.ntu.edu.tw/ cjlin/liblinear/

Methods	P@10	P@20	MAP	F1
BM25	44.6	44.1	40.8	41.5
TWE	55.4	49.5	44.2	43.5
GloVe	56.4	50.0	44.3	43.7
CBOW	56.4	49.1	44.3	43.8
Skip-gram	55.6	50.0	44.8	43.9
Jointreps	55.6	51.5	44.2	43.5
Retrofit-CBOW	57.6	50.8	44.3	43.6
Retrofit-Skipgram	56.6	50.4	44.8	43.8
SENSE-CBOW	**58.4**	**51.9**	**45.1**	**44.2**
SENSE-Skipgram	**58.2**	**50.6**	**45.0**	**44.1**

Table 3: Performance of different methods for query expansion on the RCV1 dataset. Bold scores denote that the SENSE method outperforms the corresponding baseline methods. Underlined scores are the best overall.

Conclusion and Future Work

In this paper, we proposed a novel approach for learning semantic structure-based word embedding, called SENSE. The proposed method is a jointly word embedding learning method, incorporating the corpus and the knowledge base into capturing semantics of words. Our method differs from recent related work by constructing three-level semantic structures from the KBs, and by revealing concept convergence and word divergence to unit word's semantic granularity and abstraction. Experiment results with different datasets show that the proposed method outperforms the existing state-of-the-art word embedding learning methods on various tasks.

In the future, we will study how to incorporate semantic structure information into the matrix factorization methods. We are also interested in investigating methods for effectively constructing the stable and transferable semantic structures knowledge for learning word embeddings across domains.

Acknowledgment

The research work is supported by the National Key Research and Development Program of China under Grant No.2017YFB0803302, National Nature Science Foundation of China under Grant No.61602036, and the Australian Research Council (ARC) under Discovery Grant No.DP170101632.

References

Agirre, E.; Alfonseca, E.; Hall, K.; Kravalova, J.; Paşca, M.; and Soroa, A. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *NAACL-HLT*.

Baker, S.; Reichart, R.; and Korhonen, A. 2014. An unsupervised model for instance level subcategorization acquisition. In *EMNLP*.

Barkan, O. 2017. Bayesian neural word embedding. In *AAAI*.

Bengio, Y.; Ducharme, R.; Vincent, P.; and Janvin, C. 2003. A neural probabilistic language model. *JMLR* 3:1137–1155.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *TPAMI* 35(8):1798–1828.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD*, 1247–1250.

Bollegala, D.; Maehara, T.; Yoshida, Y.; and Kawarabayashi, K. 2015. Learning word representations from relational graphs. In *AAAI*.

Bollegala, D.; Alsuhaibani, M.; Maehara, T.; and Kawarabayashi, K. 2016. Joint word representation learning using a corpus and a semantic lexicon. In *AAAI*.

Bollegala, D.; Maehara, T.; and Kawarabayashi, K. 2015. Embedding semantic relations into word representations. In *IJCAI*.

Bollegala, D.; Mu, T.; and Goulermas, J. Y. 2016. Cross-domain sentiment classification using sentiment sensitive embeddings. *TKDE* 28(2):398–410.

Bruni, E.; Boleda, G.; Baroni, M.; and Tran, N. 2012. Distributional semantics in technicolor. In *ACL*.

Cao, S., and Lu, W. 2017. Improving word embeddings with convolutional feature learning and subword information. In *AAAI*.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. P. 2011. Natural language processing (almost) from scratch. *JMLR* 12:2493–2537.

Faruqui, M.; Dodge, J.; Jauhar, S. K.; Dyer, C.; Hovy, E.; and Smith, N. A. 2015. Retrofitting word vectors to semantic lexicons. In *NAACL*.

Fellbaum, C. 1998. Wordnet: An electronic lexical database.

Ganitkevitch, J.; Durme, B. V.; and Callison-Burch, C. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, 758–764.

Goikoetxea, J.; Agirre, E.; and Soroa, A. 2016. Single or multiple? combining word representations independently learned from text and wordnet. In *AAAI*.

Johansson, R., and Piña, L. N. 2015. Embedding a semantic network in a word space. In *NAACL*.

Lampos, V.; Zou, B.; and Cox, I. J. 2017. Enhancing feature selection using word embeddings: The case of flu surveillance. In *WWW*.

Le, Q. V., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *ICML*, 1188–1196.

Lebret, R., and Collobert, R. 2014. Word embeddings through hellinger PCA. In *EACL*.

Liu, Q.; Jiang, H.; Wei, S.; Ling, Z.; and Hu, Y. 2015a. Learning semantic word embeddings based on ordinal knowledge constraints. In *ACL*.

Liu, Y.; Liu, Z.; Chua, T.; and Sun, M. 2015b. Topical word embeddings. In *AAAI*.

Luong, T.; Socher, R.; and Manning, C. D. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Miller, G. A., and Charles, W. G. 1991. Contextual correlates of semantic similarity. volume 6, 1–28.

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

Mnih, A., and Hinton, G. E. 2008. A scalable hierarchical distributed language model. In *NIPS*.

Mnih, A., and Kavukcuoglu, K. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*.

Passalis, N., and Tefas, A. 2016. Entropy optimized feature-based bag-of-words representation for information retrieval. *TKDE* 28(7):1664–1677.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Ponzetto, S. P., and Navigli, R. 2010. Knowledge-rich word sense disambiguation rivaling supervised systems. In *ACL*, 1522–1531.

Roy, D.; Ganguly, D.; Mitra, M.; and Jones, G. J. 2016. Word vector compositionality based relevance feedback using kernel density estimation. In *CIKM*.

Shen, Y.; Rong, W.; Jiang, N.; Peng, B.; Tang, J.; and Xiong, Z. 2017. Word embedding based correlation model for question/answer matching. In *AAAI*.

Spearman, C. 1904. The proof and measurement of association between two things. volume 15, 72–101.

Xu, C.; Bai, Y.; Bian, J.; Gao, B.; Wang, G.; Liu, X.; and Liu, T. 2014. RC-NET: A general framework for incorporating knowledge into word representations. In *CIKM*.

Xuan, J.; Luo, X.; Zhang, G.; Lu, J.; and Xu, Z. 2016. Uncertainty analysis for the keyword system of web events. *TSMC* 46(6):829–842.

Xuan, J.; Lu, J.; Zhang, G.; Xu, R. Y. D.; and Luo, X. 2017. Bayesian nonparametric relational topic model through dependent gamma processes. *TKDE* 29(7):1357–1369.

Yu, M., and Dredze, M. 2014. Improving lexical embeddings with semantic knowledge. In *ACL*.