A Unified Framework for Jointly Learning Distributed Representations of Word and Attributes

Liqiang Niu
niulq@nlp.nju.edu.cn

Xin-Yu Dai
daixinyu@nju.edu.cn

Shujian Huang
huangsj@nju.edu.cn

Jiajun Chen
chenjj@nju.edu.cn

National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China

Editors: Geoffrey Holmes and Tie-Yan Liu

Abstract

Distributed word representations have achieved great success in natural language processing (NLP) area. However, most distributed models focus on local context properties and learn task-specific representations individually, therefore lack the ability to fuse multi-attributes and learn jointly. In this paper, we propose a unified framework which jointly learns distributed representations of word and attributes: characteristics of word. In our models, we consider three types of attributes: topic, lemma and document. Besides learning distributed attribute representations, we find that using additional attributes is beneficial to improve word representations. Several experiments are conducted to evaluate the performance of the learned topic representations, document representations, and improved word representations, respectively. The experimental results show that our models achieve significant and competitive results.

Keywords: framework, learning, representation, word, topic, document

1. Introduction

Upon our baseline, words can be represented as indices in a vocabulary and documents can be represented as bag-of-words or bag-of-n-grams Harris (1954). Although the strategy is simple and efficient, it suffers from disadvantages such as the curse of dimensionality, data sparsity and inability to capture semantic information of words and documents.

Recently, new distributed word representations have achieved great success in many NLP applications such as POS-Tagging, Name Entity Recognition (NER), and Language Modeling Bengio et al. (2003); Collobert and Weston (2008); Turian et al. (2010). The usage of distributed representations has been extended to model concepts beyond the word level, such as phrases, sentences, documents Le and Mikolov (2014), entities, relationships Bordes et al. (2013); Socher et al. (2013), social and citation networks Tang et al. (2015). However, most models only use local context properties and learn task-specific representations individually, therefore lack the ability to fuse multi-attributes and learn jointly using both word and attributes.

In this paper, we propose a unified framework which aims at learning distributed representations of word and attributes: characteristics of word whose representations can be

Niu Dai Huang Chen

Figure 1: A toy example of the word “scoring”, where nodes represent word attributes (“football”, “score” and “David Villa opened ... There was also ... from Messi.”), and edges represent corresponding relationships (topic, lemma and document, respectively).

jointly learned with word embeddings. Naturally, word attributes correspond to syntactic relationships (POS-Tags and lemma), document structure relationships (phrase, topic and document), or other information (such as language, sentiment and name of person). For instance, as shown in Fig. 1, the word “scoring” has the following attributes: “football” (topic), “score” (lemma) and “David Villa opened ... There was also ... from Messi.” (document). Note that we can extend our models to learn more representations of attributes like “David Villa opened the scoring.” (sentence), “positive” (sentiment), “English” (language), “NN” (POS-Tag) and “Messi” (person).

Particularly, we study three kinds of attributes including topic, lemma and document. Under the unified learning framework, we propose four specific models as shown in Table 1: TW incorporates the topic attribute to learn distributed topic representations, together with learning improved word representations; DW aims at learning distributed document representations; LW incorporates the lemma attribute to improve word representations; and TLW incorporates both topic and lemma attributes to improve word representations. We summarize our contribution as follows.

- We present a unified framework for learning distributed representations of word and attributes in Section 3.1.
- Under the unified framework, our proposed models learn distributed representations of topics (TW in Section 3.2) and documents (DW in Section 3.3).
- Our proposed models (TW, LW and TLW) can improve word representations using additional attributes (topic and lemma) in Section 3.4.

The experimental results show that our models not only learn attribute representations for specific tasks, but also improve word representations using additional attributes.
2. Background: Word2Vec

Inspired by Neural Probabilistic Language Model (NPLM) Bengio et al. (2003), Mikolov et al. (2013) proposed Word2Vec for computing continuous vector representations of words from large data sets. For instance, given the word sequence \((w_{t-2}, w_{t-1}, w_{t}, w_{t+1}, w_{t+2})\), in which \(w_{t}\) is the current word, the CBOW, as shown in Fig. 2(a), predicts the word \(w_{t}\) based on the surrounding words \((w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})\), while the Skip-gram, as shown in Fig. 2(b), predicts surrounding words \((w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})\) given the current word \(w_{t}\).

When training, given a word sequence \(D = \{w_{1}, ..., w_{M}\}\), the learning objective functions are defined to maximize the following log-likelihoods, based on CBOW and Skip-gram, respectively.

\[
\mathcal{L}_{CBOW}(D) = \frac{1}{M} \sum_{i=1}^{M} \log p(w_{i}|w_{\text{ctx}}), \tag{1a}
\]

\[
\mathcal{L}_{Skip\text{-gram}}(D) = \frac{1}{M} \sum_{i=1}^{M} \sum_{-k \leq c \leq k, c \neq 0} \log p(w_{i+c}|w_{i}). \tag{1b}
\]

Here, in Equation (1a), \(w_{\text{ctx}}\) indicates the context of the current word \(w_{i}\). In Equation (1b), \(k\) is the window size of context. For any variables \(w_{j}\) and \(w_{i}\), the conditional probability \(p(w_{j}|w_{i})\) is calculated using softmax function as follows,

\[
p(w_{j}|w_{i}) = \frac{\exp(w_{j} \cdot w_{i})}{\sum_{w \in W} \exp(w \cdot w_{i})}, \tag{2}
\]

where \(w, w_{i}\) and \(w_{j}\) are respectively the word representations of word \(w, w_{i}\) and \(w_{j}\), \(W\) is the word vocabulary.

3. Our Models

3.1. A Unified Framework

Inspired by NPLM and Word2Vec, as shown in Fig. 2 (c) and (d), we propose a unified framework for distributed representations of word and attributes: characteristics of word whose representations can be jointly learned with word embeddings. For instance, given a word sequence \((w_{t-2}, w_{t-1}, w_{t}, w_{t+1}, w_{t+2})\) in which \(w_{t}\) is the current word assigned with \(k\) attributes \((a_{t,1}, ..., a_{t,k})\), the CBOW, as shown in Fig. 2(c), predicts the word \(w_{t}\) and \(k\) attributes \((a_{t,1}, ..., a_{t,k})\) based on the surrounding words \((w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})\), while the
### 3.2. TW: Learning Topic Representations

As shown in Table 1, TW considers the topic attribute assigned to word and aims at learning distributed topic representations. For instance, given a word-topic sequence \((w_{t-2}, z_{t-2} : w_{t-1}, z_{t-1} : w_t, z_t : w_{t+1}, z_{t+1} : w_{t+2}, z_{t+2})\), in which \(w_t\) is the current word paired with a topic attribute \(z_t\) learned from \textit{GibbsLDA++}\(^1\), the CBOW predicts the word \(w_t\).

---


---

Figure 2: The CBOW and Skip-gram architectures of Word2Vec and our framework, where \((w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})\) are context words and \(w_t\) is current word paired with \(k\) attributes \((a_{t1}, ..., a_{tk})\).
and topic $z_t$ based on the surrounding words $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$, while the Skip-gram predicts surrounding words $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ given the current word $w_t$ and topic $z_t$.

When training, given a word-topic sequence $D = \{w_1 : z_1, ..., w_M : z_M\}$, the learning objective functions can be defined to maximize the following log-likelihoods, based on CBOW and Skip-gram, respectively.

$$\mathcal{L}_{CBOW}(D) = \frac{1}{M} \sum_{i=1}^{M} \left( \log p(w_i | w_{\text{cxt}}) + \log p(z_i | w_{\text{cxt}}) \right),$$  \hspace{1cm} (3a) \\
$$\mathcal{L}_{\text{Skip-gram}}(D) = \frac{1}{M} \sum_{i=1}^{M} \sum_{-k \leq c \leq k, c \neq 0} \left( \log p(w_{i+c} | w_i) + \log p(w_{i+c} | z_i) \right).$$  \hspace{1cm} (3b)$$

Note that, in Equation (3a) and (3b), the first part about $w_i$ is based on Hypothesis A and the second part about $z_i$ is based on Hypothesis B. Different with traditional topic as a probability distribution over words in LDA, TW embeds words and topics in the same semantic space in which similarity can be measured immediately by cosine function.

### 3.3. DW: Learning Document Representations

As shown in Table 1, DW considers the document attribute assigned to word and aims at learning distributed document representations. For instance, given a word-document sequence $(w_{t-2} : d_{t-2}, w_{t-1} : d_{t-1}, w_t : d_t, w_{t+1} : d_{t+1}, w_{t+2} : d_{t+2})$ in which $w_t$ is the current word paired with a document attribute $d_t$, the CBOW predicts the word $w_t$ and document $d_t$ based on the surrounding words $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$, while the Skip-gram predicts surrounding words $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ given the current word $w_t$ and document $d_t$.

When training, given a word-document sequence $D = \{w_1 : d_1, ..., w_M : d_M\}$, the learning objective functions can be defined to maximize the following log-likelihoods, based on CBOW and Skip-gram, respectively.

$$\mathcal{L}_{CBOW}(D) = \frac{1}{M} \sum_{i=1}^{M} \left( \log p(w_i | w_{\text{cxt}}) + \log p(d_i | w_{\text{cxt}}) \right),$$  \hspace{1cm} (4a) \\
$$\mathcal{L}_{\text{Skip-gram}}(D) = \frac{1}{M} \sum_{i=1}^{M} \sum_{-k \leq c \leq k, c \neq 0} \left( \log p(w_{i+c} | w_i) + \log p(w_{i+c} | d_i) \right).$$  \hspace{1cm} (4b)$$

Note that, in Equation (4a) and (4b), the first part about $w_i$ is based on Hypothesis A and the second part about $d_i$ is based on Hypothesis D. Document as a attribute leads to that all words in the same document are less distinguishable, and then DW makes word representations worse. So in this paper, DW only focuses on learning distributed document representations without improving word representations.

### 3.4. Improving Word Representations

**TW** As described previously in Section 3.2, TW learns distributed topic representations jointly with word representations. Comparing to Word2Vec which only uses local context words, TW takes into account both word and topic attribute. Naturally, we desire that using additional topic can improve original word representations.
lemma: "go" 

word: "goes" 

word: "going" 

word: "went"

Figure 3: An example of lemma and variational words in morphology.

**LW** In morphology, a lemma\(^2\) is the canonical form of a set of words. In English, for example, "go", "goes", "went" and "going" are forms of the same lexeme, with "go" as the lemma (show in Fig. 3). Different words with the same lemma usually contain the same basic meanings.

As shown in Table 1, LW considers the lemma attribute paired with word and aims at improving word representations. For instance, given a word-lemma sequence \((w_{t-2}: l_{t-2}, w_{t-1}: l_{t-1}, w_t: l_t, w_{t+1}: l_{t+1}, w_{t+2}: l_{t+2})\), in which \(w_t\) is the current word paired with a lemma attribute \(l_t\) obtained from WordNet Lemmatizer\(^3\), the CBOW predicts the word \(w_t\) and lemma \(l_t\) based on the surrounding words \((w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})\), while the Skip-gram predicts surrounding words \((w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})\) given the current word \(w_t\) and lemma \(l_t\).

When training, given a word-lemma sequence \(D = \{w_1 : l_1, ..., w_M : l_M\}\), the learning objective functions can be defined to maximize the following log-likelihoods, based on CBOW and Skip-gram, respectively.

\[
\mathcal{L}_{\text{CBOW}}(D) = \frac{1}{M} \sum_{i=1}^{M} (\log p(w_i | w_{\text{ctx}}) + \log p(l_t | w_{\text{ctx}})),
\]

\[
\mathcal{L}_{\text{Skip-gram}}(D) = \frac{1}{M} \sum_{i=1}^{M} \sum_{-k \leq c \leq k, c \neq 0} (\log p(w_{i+c} | w_i) + \log p(w_{i+c} | l_i)).
\]

Note that, in Equation (5a) and (5b), the first part about \(w_t\) is based on **Hypothesis A** and the second part about \(l_t\) is based on **Hypothesis C**.

**TLW** As shown in Table 1, TLW considers both topic and lemma attributes and aims at improving word representations. For instance, given a word-topic-lemma sequence \((w_{t-2}: z_{t-2}, l_{t-2}, w_{t-1}: z_{t-1}, l_{t-1}, w_t: z_t, l_t, w_{t+1}: z_{t+1}, l_{t+1}, w_{t+2}: z_{t+2}, l_{t+2})\), in which \(w_t\) is the current word paired with a topic \(z_t\) and a lemma \(l_t\), the CBOW predicts the word \(w_t\), topic \(z_t\) and lemma \(l_t\) based on the surrounding words \((w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})\), while the Skip-gram predicts surrounding words \((w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})\) given the current word \(w_t\), topic \(z_t\) and lemma \(l_t\).

When training, given a word-topic-lemma sequence \(D = \{w_1 : z_1 : l_1, ..., w_M : z_M : l_M\}\), the learning objective functions can be defined to maximize the following log-likelihoods, based on CBOW and Skip-gram, respectively.

\(^2\) http://en.wikipedia.org/wiki/Lemma

\(^3\) http://textanalysisonline.com/nltk-wordnet-lemmatizer

148
Learning Distributed Representations of Word and Attributes

\[ \mathcal{L}_{CBOW}(D) = \frac{1}{M} \sum_{i=1}^{M} \left( \log p(w_i|w_{ext}) + \log p(z_i|w_{ext}) + \log p(l_i|w_{ext}) \right), \]

\[ \mathcal{L}_{Skip-gram}(D) = \frac{1}{M} \sum_{i=1}^{M} \sum_{-k \leq c \leq k, c \neq 0} \left( \log p(w_{i+c}|w_i) + \log p(w_{i+c}|z_i) + \log p(w_{i+c}|l_i) \right). \]

Note that, in Equation (6a) and (6b), the first part about \( w_i \) is based on Hypothesis A, the second part about \( z_i \) is based on Hypothesis B and the third part about \( l_i \) is based on Hypothesis C.

3.5. Optimization and Learning Process

We follow the optimization scheme that used in Word2Vec. To approximately maximize the log probability of the softmax, we use Negative Sampling without Hierarchical Softmax Mikolov et al. (2013b). Stochastic gradient descent (SGD) and back-propagation algorithm are used to optimize our models.

In particular, TW focuses on learning topic representations, while DW turns to learn document representations. Moreover, TW, LW and TLW can improve word representations. In our implementations, these models of TW, LW and TLW firstly initialize the word and topic representations randomly, and then learn their distributed representations jointly. When DW, we only initialize document vectors randomly and initialize word vectors using pre-trained word representations learned from large and high-quality datasets. And then DW learn distributed document representations while keeping word representations unchanged. In general, the complexity of our models is linear with the size of dataset, same with Word2Vec Mikolov et al. (2013).

4. Experiments

4.1. Datasets

We use the English Gigaword 4 as our training data for learning fundamental word representations. Actually, we randomly choose some documents and constructed two different sized training sets as described in the following:

- **DS-100k**: we choose 100,000 documents, each consisting of more than 1,000 characters from subfolder ltw_eng (Los Angeles Times) which contains 411,032 documents. Besides, we eliminate those words that occur less than 5 times and the stop words. In the end, DS-100k contains about 42 million words and the vocabulary size is 102,644.

- **DS-500k**: we also choose 500,000 documents similarly from subfolder nyt_eng (New York Times) which contains 1,962,178 documents. After eliminating these words that occur less than 5 times and the stop words, DS-500k finally contains about 0.21 billion words and the vocabulary size is 232,481.

Besides, we run GibbsLDA++ and TW on DS-100k for topic evaluation and run DW on 20NewsGroup5 for document evaluation.

---

Blei et al. (2003). We clustered words into topics as follows: first perform experiments to evaluate our topic representations compared to original LDA. TW learns distributed topic representations together with word representations. So, we can immediately calculate the cosine similarity between words and topics. For each topic, we select higher similarity words.

4.2. Evaluation for Topic Representations

TW learns distributed topic representations together with word representations. So, we first perform experiments to evaluate our topic representations compared to original LDA Blei et al. (2003). We clustered words into topics as follows:

- LDA: all topics are represented as probability distribution over words. We select the top $N = 15$ words with highest probability conditioned on the selected topic.

- TW: all topics and words are equally represented as the low-dimensional dense vectors, we can immediately calculate the cosine similarity between words and topics. For each topic, we select higher similarity words.

Fig. 4(a) shows the top 15 nearest words clustered from LDA and TW for some selected topics, respectively. We now give more detailed analysis to understand the difference between them. As shown in Fig. 4(a), in Topic_19, LDA gives the words like “drug”, “drugs”, “cancer” and “patients”, while TW returns “aricept”, “memantine”, “enbrel” and “gabapentin”. In Topic_27, LDA gives the words of “medical”, “hospital”, “care”, “patients”.

Table: Nearest words and topics for each topic.

<table>
<thead>
<tr>
<th>Topic_6</th>
<th>Topic_19</th>
<th>Topic_27</th>
<th>Topic_79</th>
</tr>
</thead>
<tbody>
<tr>
<td>food</td>
<td>drug</td>
<td>medical</td>
<td>computer</td>
</tr>
<tr>
<td>restaurant</td>
<td>drugs</td>
<td>hospital</td>
<td>technology</td>
</tr>
<tr>
<td>eat</td>
<td>cancer</td>
<td>care</td>
<td>phone</td>
</tr>
</tbody>
</table>

Figure 4: (a): Nearest words and topics for each topic. Words are listed with corresponding probabilities in LDA while words and topics are listed with calculated cosine similarity in TW. (b): t-SNE 2D embedding of the nearest word representation for each topic in LDA (above) and TW (below).
### Table 2: The performance of DW compared to other approaches on 20NewsGroup. The results of other methods are reported in Liu et al. (2015). Bold scores are the best overall related models.

<table>
<thead>
<tr>
<th>Models</th>
<th>Dim</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>80</td>
<td>72.2</td>
<td>70.8</td>
<td>70.7</td>
<td>70.9</td>
</tr>
<tr>
<td>PV-DM</td>
<td>400</td>
<td>72.4</td>
<td>72.1</td>
<td>71.5</td>
<td>71.5</td>
</tr>
<tr>
<td>PV-DBOW</td>
<td>400</td>
<td>75.4</td>
<td>74.9</td>
<td>74.3</td>
<td>74.3</td>
</tr>
<tr>
<td>DW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBOW</td>
<td>200</td>
<td>75.4</td>
<td>74.9</td>
<td>74.3</td>
<td>74.3</td>
</tr>
<tr>
<td>Skip-gram</td>
<td>400</td>
<td>75.8</td>
<td>74.4</td>
<td>74.9</td>
<td>74.8</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>72.1</td>
<td>71.6</td>
<td>71.1</td>
<td>71.1</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>72.5</td>
<td>72.1</td>
<td>72.1</td>
<td>72.1</td>
</tr>
</tbody>
</table>

and “doctors”, while TW returns “neonatal”, “anesthesiologists”, “anesthesia” and “comatose”. We only know that Topic 19 and Topic 27 share the same topic about “patients” or “medical”, but we can’t get their difference from the results of LDA. But from the result of TW, we can easily discover that Topic 19 focuses on a more specific topic about drugs (“aricept”, “memantine”, “enbrel” and “gabapentin”), while Topic 27 focuses on another specific topic about treatment (“anesthesiologists”, “anesthesia” and “comatose”), they are absolutely different. Obviously, TW presents more distinguished results between two similar topics.

Fig. 4(b) shows the 2D embedding of the corresponding related words for each topic by using t-SNE. Obviously, TW produces a better grouping and separation of the words in different topics. In contrast, LDA does not produce a well separated embedding, and words in different topics tend to mix together.

In summary, for each topic, words selected by TW are more typical and representative compared to those returned by LDA. Eventually, TW can do better at distinguishing different topics.

Note that TLW can generate the similar results of topics as TW, we don’t show them due to the space limitation.

### 4.3. Evaluation for Document Representations

#### Text Classification

DW focuses on learning distributed document representations and we perform a multi-class text classification task to evaluate it. We use the standard dataset 20NewsGroup which consists of about 20,000 documents collected from 20 different newsgroups. Considering the insufficient training data of 20NewsGroup, we firstly learn word representations from large dataset DS-500k. Then DW starts learning distributed document representations while keeping word representations unchanged.

For each document, DW returns a corresponding vector as its representation. And then we deploy LIBLINEAR\(^6\) which uses the “one vs rest” method for multi-category classification. For evaluating the effectiveness of our models, we compare DW with another document representation models including LDA and recently proposed Paragraph Vector models Le and Mikolov (2014). LDA represents each document as a probability distribution over latent topics, while Paragraph Vector models represent each document as a low-dimensional dense vector, including the distributed memory model (PV-DM) and the distributed bag-of-words model (PV-DBOW). Table 2 shows that DW achieves competitive

\(^6\) http://www.csie.ntu.edu.tw/ cjlin/liblinear/
Table 3: Accuracy (%) in word analogy tasks, higher values are better. We compare our models (TW, LW and TLW) with baseline model W2V (Word2Vec) and state-of-the-art Glove. Bold scores are the best of our models for each dataset. Time is roughly estimated on a single machine with 8GB RAM.

4.4. Evaluation for Improved Word Representations

Finally we evaluate the improved word representations in the following benchmark tasks. **Word analogy** Two datasets are used for this task. The Google dataset proposed by Mikolov et al. (2013) contains 10,675 syntactic questions (e.g., young:yonger::large:larger) and 8,869 semantic questions (e.g., Rome:Italy::Athens:Greece). The MSR dataset proposed by Mikolov et al. (2013c) contains 8,000 syntactic questions (e.g., good:better::rough:rougher). In each question, the fourth word is missing, and the task is to correctly predict the fourth word. We use the vector offset method Mikolov et al. (2013b) to compute the vector $w_{\text{fourth}} = w_{\text{third}} + (w_{\text{second}} - w_{\text{first}})$, if the vector $w_{\text{fourth}}$ has the highest cosine similarity with the correct answer, this question is correctly answered.

We compare the results of our models with the baseline Word2Vec Mikolov et al. (2013) and state-of-the-art Glove Pennington et al. (2014). As shown in Table 3, LW and TLW present better performance than Word2Vec in most Skip-gram cases while TW does not. It seems that lemma knowledge can get more improvement than topic in word analogy tasks. More accurately, on DS-100K, TLW improves +8.48% on Google semantic while LW improves +5.57% on MSR syntactic in Skip-gram. On bigger DS-500K, LW improves +3.95% on Google semantic and +2.72% on MSR syntactic in Skip-gram. In general, we have the following conclusions:

- Using additional lemma leads to better word representations in word analogy tasks.

*http://research.microsoft.com/enus/projects/rnn/default.aspx*
*http://nlp.stanford.edu/projects/glove/*

Table 4: Comparing Spearman rank correlation coefficient of our models (TW, LW and TLW) with Word2Vec and Glove on WordSim-353. Bold scores are the best overall for each dataset.

- Using additional lemma can achieve significant improvement on small datasets. When dataset becoming larger, extra information will help less. The result is also consistent with the saying that “More data usually beats better algorithms” Rajaraman (2008).

Note that both Word2Vec and our models perform worse than Glove, which trains on global word-word co-occurrence counts rather than local context windows used in Word2Vec and our models. So we perform more experiments to further compare these models.

Word similarity Next we further perform the second task of word similarity on another WordSim-353 dataset Finkelstein et al. (2001) to prove the effectiveness of our models and we consistently compare our models with Word2Vec and Glove. In Table 4, our models achieve significant improvement on word similarity compared to Word2Vec and also perform a lot better than Glove.

Now using additional topic and lemma knowledge can improve original word representations significantly, especially in small datasets. In general, we have the idea that using extra knowledge can alleviate the shortage of datasets in some specific domains.

5. Conclusion and Future Work

In this paper, we propose a unified framework for learning distributed representations of word and attributes. In particular, we consider topic, lemma and document attributes and present four specific models (TW, DW, LW and TLW), respectively. From the observation and analysis in experiments, our models not only learn topic and document representations which achieve distinct and competitive results in corresponding tasks, but also improve original word representations significantly.

Finally we want to emphasize that our proposed framework is flexible and scalable to incorporate more attributes. In the future, we will explore the usage of other word attributes, such as sentiment for sentiment analysis, POS-Tags for POS-Tagging and name of person for NER. Besides, we will exploit new methods which can simultaneously infer topic for each word and learn topic embeddings without using LDA previously.
Acknowledgments

This work was supported by the NSFC (61472183,61333014) and the 863 High-Tech Program (2015AA015406).

References


Eric H. Huang, Richard Socher, Christopher D. Manning and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Annual Meeting of the Association for Computational Linguistics*. 154


Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua and Maosong Sun. 2015. Topical word embeddings. In Association for the Advancement of Artificial Intelligence.


