JMLR: Workshop and Conference Proceedings 45:143–156, 2015 ACML 2015

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 Techonology
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

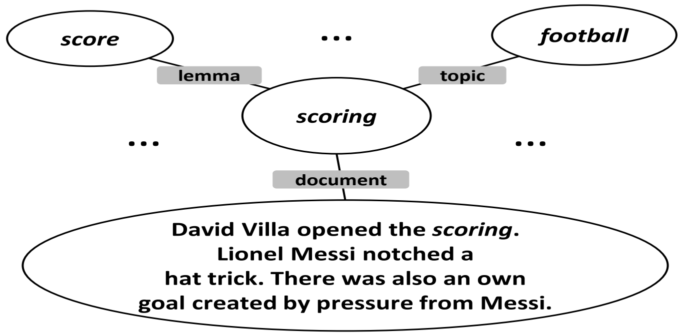


Figure 1: A toy example of the word "*scoring*", where nodes represent word attributes ("*football*", "*score*" and "*David Villa opened ... There was also ... from Messi.*"), and edges represent corresponding relationships (topic, lemma and document, respectively).

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.

Models	Word and Attributes	Learning Targets
Word2Vec	word	word representations
TW	word:topic	topic representations and improved word representations
DW	word:document	document representations
LW	word:lemma	improved word representations
TLW	word:topic:lemma	improved word representations

Table 1: Pairs of word and attributes and learning targets used in Word2Vec Mikolov et al. (2013) and our models (TW, DW, LW and TLW).

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.

$$\mathcal{L}_{CBOW}(D) = \frac{1}{M} \sum_{i=1}^{M} \log p(w_i|w_{cxt}), \tag{1a}$$

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

Here, in Equation (1a), w_{cxt} 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,

$$p(w_j|w_i) = \frac{\exp(\mathbf{w_j} \cdot \mathbf{w_i})}{\sum_{w \in W} \exp(\mathbf{w} \cdot \mathbf{w_i})}, \tag{2}$$

where $\mathbf{w}$, $\mathbf{w_i}$ and $\mathbf{w_j}$ 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

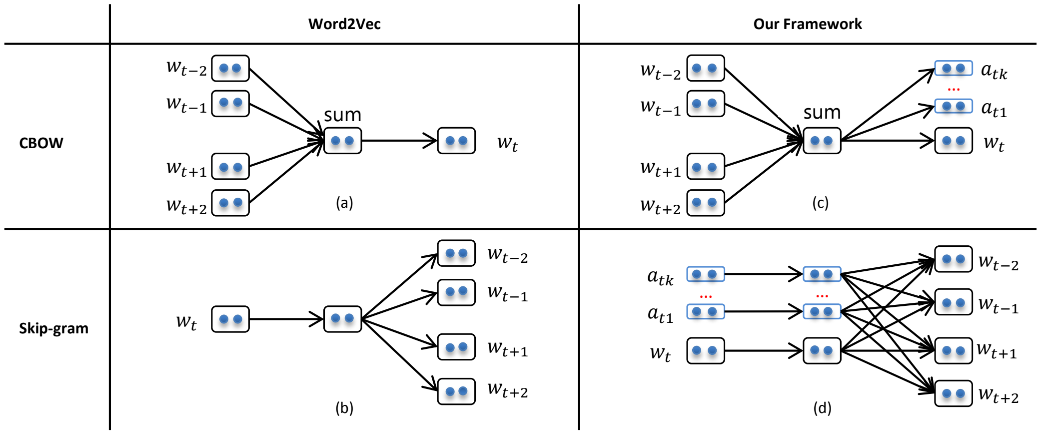


Figure 2: The CBOW and Skip-gram architectures of Word2Vec and our framework, where $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ are context words and w_t is current word paired with k attributes $(a_{t1}, ..., a_{tk})$.

Skip-gram, as shown in Fig. 2(d), predicts surrounding words $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ given the current word w_t and k attributes $(a_{t,1}, ..., a_{t,k})$.

Based on our proposed framework, it is obviously seen that words and attributes can help with each other for better representations during the learning process.

In this paper, we consider three kinds of attributes: topic, lemma and document, and we propose our models of TW, DW, LW and TLW, respectively. Inspired by word distributional hypothesis, we assume that word attributes have the similar distributional hypotheses. Moreover, our models are also motivated by following distributional hypotheses:

- **Hypothesis A**: *"words that occur in the same contexts tend to have similar meanings" (Pantel, 2005).*
- **Hypothesis B**: *"topics assigned to words that occur in the same contexts tend to be similar".*
- **Hypothesis C**: *"lemmas of words that occur in the same contexts tend to be similar".*
- **Hypothesis D**: *"documents consisting of words that occur in the same contexts tend to be similar".*

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 *GibbsLDA++*[1], the CBOW predicts the word w_t

1. http://gibbslda.sourceforge.net/

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.

$$\mathcal{L}_{CBOW}(D) = \frac{1}{M}\sum_{i=1}^{M}(\log p(w_i|w_{cxt}) + \log p(z_i|w_{cxt})), \tag{3a}$$

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

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 immedidately 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.

$$\mathcal{L}_{CBOW}(D) = \frac{1}{M}\sum_{i=1}^{M}(\log p(w_i|w_{cxt}) + \log p(d_i|w_{cxt})), \tag{4a}$$

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

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.

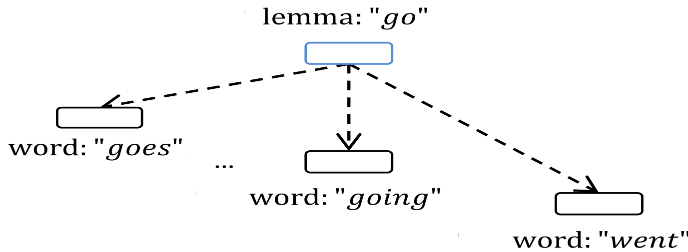


Figure 3: An example of lemma and variational words in morphology.

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.

$$\mathcal{L}_{CBOW}(D) = \frac{1}{M} \sum_{i=1}^{M} (\log p(w_i|w_{cxt}) + \log p(l_i|w_{cxt})), \tag{5a}$$

$$\mathcal{L}_{Skip-gram}(D) = \frac{1}{M} \sum_{i=1}^{M} \sum_{-k \leq c \leq k, c \neq 0} (\log p(w_{i+c}|w_i) + \log p(w_{i+c}|l_i)). \tag{5b}$$

Note that, in Equation (5a) and (5b), the first part about w_i is based on **Hypothesis A** and the second part about l_i 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

$$\mathcal{L}_{CBOW}(D) = \frac{1}{M}\sum_{i=1}^{M}(\log p(w_i|w_{cxt}) + \log p(z_i|w_{cxt}) + \log p(l_i|w_{cxt})), \tag{6a}$$

$$\mathcal{L}_{Skip-gram}(D) = \frac{1}{M}\sum_{i=1}^{M}\sum_{-k\leq c\leq k, c\neq 0}(\log p(w_{i+c}|w_i) + \log p(w_{i+c}|z_i) + \log p(w_{i+c}|l_i)). \tag{6b}$$

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:

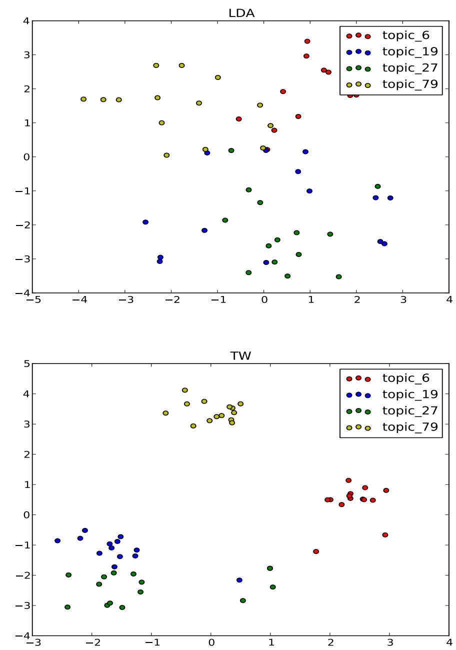
- **DS-100k**: we choose 100,000 documents, each consisting of more than 1,000 characters from subfolder ltw_eng (Los Angeles Times) which contains 411,032 documents. Besides, we eliminate those words that occur less than 5 times and the stop words. In the end, DS-100k contains about 42 million words 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 20NewsGroup[5] for document evaluation.

4. https://catalog.ldc.upenn.edu/LDC2011T07
5. http://qwone.com/ jason/20Newsgroups/

	Topic_6		Topic_19		Topic_27		Topic_79	
	word	prob.	word	prob.	word	prob.	word	prob.
LDA	food	0.027	drug	0.031	medical	0.033	computer	0.016
	restaurant	0.008	drugs	0.019	hospital	0.024	technology	0.010
	eat	0.008	cancer	0.019	care	0.019	phone	0.009
	more	0.005	study	0.011	patients	0.018	software	0.009
	chicken	0.005	patients	0.011	doctors	0.016	digital	0.008
	cooking	0.005	treatment	0.009	health	0.013	apple	0.008
	eating	0.005	fda	0.009	doctor	0.009	use	0.007
	one	0.005	heart	0.008	patient	0.009	system	0.006
	good	0.005	risk	0.008	surgery	0.008	microsoft	0.006
	foods	0.005	more	0.007	center	0.008	up	0.006
	dinner	0.004	use	0.007	treatment	0.007	music	0.006
	make	0.004	blood	0.007	hospitals	0.007	video	0.006
	fresh	0.004	women	0.006	heart	0.006	one	0.006
	chef	0.004	disease	0.006	dr	0.006	more	0.005
	made	0.004	percent	0.005	one	0.005	computers	0.005
	word/topic	cos.	word/topic	cos.	word/topic	cos.	word/topic	cos.
TW	cheeseburgers	0.564	topic_62	0.618	topic_19	0.519	wirelessly	0.584
	meatless	0.535	aricept	0.531	topic_62	0.478	handhelds	0.573
	smoothies	0.534	topic_27	0.519	neonatal	0.466	desktops	0.572
	topic_95	0.533	memantine	0.514	topic_13	0.457	pda	0.566
	meatloaf	0.530	enbrel	0.512	anesthesiologists	0.445	smartphone	0.566
	tastier	0.530	gabapentin	0.511	anesthesia	0.439	megabyte	0.562
	topic_52	0.527	colorectal	0.509	reconstructive	0.437	macbook	0.556
	cheeseburger	0.525	prilosec	0.507	comatose	0.437	handheld	0.549
	concoctions	0.522	placebos	0.507	hysterectomy	0.433	treo	0.549
	vegetarians	0.515	intravenously	0.504	ventilator	0.432	modems	0.548
	twinkies	0.514	adderall	0.502	checkup	0.429	camcorders	0.547
	veggie	0.513	inhibitor	0.502	pacemaker	0.428	toshiba	0.545
	panera	0.513	opioid	0.501	aneurysms	0.423	peripherals	0.545
	pepperoni	0.507	oncologists	0.501	respirator	0.423	android	0.544
	condiments	0.504	precancerous	0.501	caesarean	0.422	centrino	0.543



(a) Nearest words and topics

(b) t-SNE 2D embedding

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).

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*"

Models		Dim	Accuracy	Precision	Recall	F1-Measure
LDA		80	72.2	70.8	70.7	70.0
PV-DM		400	72.4	72.1	71.5	71.5
PV-DBOW		400	75.4	74.9	74.3	74.3
DW	CBOW	300	74.4	73.9	73.5	73.4
		400	**75.8**	**75.4**	**74.9**	**74.8**
	Skip-gram	300	72.1	71.5	71.2	71.1
		400	72.9	72.4	72.1	72.2

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.

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/

Models (dim=300)		Dataset	Google			MSR	Time
			semantic	syntactic	total	syntactic	hours
CBOW	W2V	DS-100k	19.08	33.73	27.69	32.36	0.1
	TW	DS-100k	20.42	31.42	26.88	31.47	0.2
	LW	DS-100k	28.64	25.71	26.92	29.35	0.2
	TLW	DS-100k	28.15	27.32	27.67	30.21	0.2
Skip-gram	W2V	DS-100k	27.56	35.63	32.31	29.85	1.1
	TW	DS-100k	31.26	35.13	33.53	29.03	1.2
	LW	DS-100k	33.94	**37.13**(+1.50)	36.16	**35.42**(+5.57)	1.2
	TLW	DS-100k	**36.04**(+8.48)	36.60	**36.37**(+4.06)	34.65	1.3
Glove:iter=5		DS-100k	43.64	40.83	41.99	39.47	1.1
CBOW	W2V	DS-500k	30.57	50.57	41.74	44.97	2.1
	TW	DS-500k	28.12	49.60	40.12	43.93	2.2
	LW	DS-500k	41.80	46.11	44.21	42.43	2.2
	TLW	DS-500k	41.76	47.63	45.04	44.44	2.2
Skip-gram	W2V	DS-500k	41.77	50.63	46.89	43.38	6.8
	TW	DS-500k	41.46	49.46	45.93	41.39	7.4
	LW	DS-500k	**45.72**(+3.95)	**50.86**(+0.23)	**48.59**(+1.7)	**46.10**(+2.72)	7.2
	TLW	DS-500k	44.85	50.58	48.05	45.62	7.7
Glove:iter=5		DS-500k	51.32	49.12	50.09	46.36	6.3
Glove:iter=15		DS-500k	51.88	53.41	52.74	48.32	17.2

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.

results with existing models. Note that all these models did not perform better than BOW and TWE-1 reported in Liu et al. (2015) which both use additional TF-IDF feature to help classification.

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[7] 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 $\mathbf{w_{fourth}} = \mathbf{w_{third}} + (\mathbf{w_{second}} - \mathbf{w_{first}})$, if the vector $\mathbf{w_{fourth}}$ 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[8] 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.

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

Model (dim=300)		**Corpus**	$\rho \times 100$
Glove:iter=5		DS-100k	51.9
CBOW	Word2Vec	DS-100k	55.6
	TW	DS-100k	62.6
	LW	DS-100k	63.9
	TLW	DS-100k	65.0
Skip-gram	Word2Vec	DS-100k	61.5
	TW	DS-100k	63.7
	LW	DS-100k	**65.4**
	TLW	DS-100k	63.5
Glove:iter=5		DS-500k	50.8
Glove:iter=15		DS-500k	50.9
CBOW	Word2Vec	DS-500k	63.7
	TW	DS-500k	62.2
	LW	DS-500k	65.9
	TLW	DS-500k	**67.5**
Skip-gram	Word2Vec	DS-500k	65.8
	TW	DS-500k	63.7
	LW	DS-500k	64.6
	TLW	DS-500k	63.9

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

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

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine learning*, ICML '08, pages 160-167, New York, NY, USA. ACM.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. Discourse processes 25.2-3 (1998): 259-284.

David M. Blei, Andrew Y. Ng and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993-1022.

Tomas Mikolov, Kai Chen, Gerg Corrado and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111-3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746-751, Atlanta, Georgia, June. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning*, Beijing, China.

Richard Socher, Eric H. Huang, Jeffrey Penniington, Andrew Y. Ng and Christopher D. Manning. 2011a. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems 24*.

Richard Socher, Cliff C. Lin, Andrew Y. Ng and Christopher D. Manning. 2011b. Parsing natural scenes and natural language with recursive neural network. In *Proceedings of the 26th International Conference on Machine Learning*.

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*.

Jeffrey Pennington, Richard Socher and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Conference on Empirical Methods in Natural Language Processing*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston and Oksana Yakhenko. 2013. Translating Embeddings for Modeling Multi-relational data. In *NIPS*, pages 2787-2795.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with Neural Tensor Networks for Knowledge Base Completion. In *NIPS*, pages 926-934.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan and Qiaozhu Mei. 2015. LINE:Large-scale Information Network Embeddings. In *WWW 2015*, May 18-22, 2015, Florence, Italy.

Zellig S. Harris. 1954. Distributional structure. *Word*.

Andrew L. Mass, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Survey*, 34(1): 1-47.

Dani Yogatama, Manaal Faruqui. Chris Dyer and Noah A. Smith. 2014. Learning word representations with hierarchical sparse coding. In *NIPS Deep Learning and Representation Learning Workshop*, Montréal, Quebec, December 2014.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*.

Andriy Mnih and Geoffrey E. Hinton. 2009. A scalable hierarchical distributed language model. In *Advances in Neural Information Processing Systems*, pages 1081-1088.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, pages 246-252.

Joseph Turian, Lev Ratinov and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384-394. Association for Computational Linguistics.

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*.

Patrick Pantel and Marina del Rey. 2005. Inducing ontological co-occurrence vectors. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 125-132.

Magnus Sahlgren. 2008. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1): 33-54.

Lev Finkelstein, Evgeniy Gabrilocivh, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *Proceedings of the 10th International Conference on Word Wide Web*, WWW '01, pages 406-414, New York, NY, USA. ACM.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546-556.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North America Chapter of the Association for Computational Linguistics*, HLT '10, pages 109-117, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nitin Madnani and Joel Tetreault. 2012. Re-examining Machine Translation Metrics for Paraphrase Identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182-190, Montréal, Canada, June. Association for Computational Linguistics.

Bill Dolan, Chris Quirk and Chris Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *COLING*, 2004.

Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao and Tie-Yan Liu. 2014. Co-learning of Word Representations and Morpheme Representations. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 141-150, Dublin, Ireland, August 23-29 2014.

Anand Rajaraman. 2008. More data usually beats better algorithms. *Datawocky Blog.*

Nal Kalchbrenner, Edward Grefenstette and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.*

Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov. 2014. A Multiplicative Model for Learning Distributed Text-Based Attribute Representations. In *Neural Information Processing Systems (NIPS)*, Montreal, Canada, December 2014.

Van der Maaten, Laurens, and Geoffrey Hinton. 2008. Visualizig data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605): 85.