This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2632521, IEEE/ACM Transactions on Audio, Speech, and Language Processing

Topic-Aware Deep Compositional Models for Sentence Classification

Rui Zhao and Kezhi Mao

***Abstract*—In recent years, deep compositional models have emerged as a popular technique for representation learning of sentence in computational linguistic and natural language processing. These models normally train various forms of neural networks on top of pre-trained word embeddings using a task-specific corpus. However, most of these works neglect the multi-sense nature of words in the pre-trained word embeddings. In this paper we introduce topic models to enrich the word embeddings for multi-senses of words. The integration of the topic model with various semantic compositional processes leads to Topic-Aware Convolutional Neural Network (TopCNN) and Topic-Aware Long Short Term Memory Networks (TopLSTMs). Different from previous multi-sense word embeddings models that assign multiple independent and sense-specific embeddings to each word, our proposed models are lightweight and flexible frameworks that regard word sense as the composition of two parts: a general sense derived from a large corpus and a topic-specific sense derived from a task-specific corpus. In addition, our proposed models focus on semantic composition instead of word understanding. With the help of topic models, we can integrate the topic-specific sense at word-level before composition and sentence-level after composition. Comprehensive experiments on five public sentence classification datasets are conducted, and the results show that our proposed Topic-Aware deep compositional models produce competitive or better performance than other text representation learning methods.**

***Index Terms*—Machine Learning, natural language processing, sentence classification, text representation learning.**

I. Introduction

Sentence modeling and classification is a keystone of various natural language processing (NLP) tasks, including sentiment analysis, spam detection, paraphrase detection, machine translation and so on [1], [2], [3]. Modeling of sentence aims to capture and analyze the semantic information of a sentence. In this problem, the core and also critical step is the representation learning, which aims to learn a fixed-length numerical vector for a sentence. Classical representation models include Bag-of-Words (BoW) models and topic models.

With the development of deep learning methods in the last few years, representation learning for different levels of text units has been redefined [4], [5], [6]. In deep learning models for NLP, the compositional models are built on word embeddings as shown in Figure 1. In contrast to one-hot word representation which is a extreme sparse 1-of-V vector (V is the vocabulary size), word embeddings encode semantic and syntactic information of words into a low-dimensional and dense representation through neural language models trained on a large-scale text corpus without annotations [7], [8], [9]. Another important ingredient of deep learning models is the applied compositional models. Before deep learning models become popular, the commonly used models are shallow models based on fixed algebraic operations including addition, tensor production and so on [10]. Deep models mainly adopt multi-layer neural networks including recursive [11], [12], recurrent [13], [14] and convolutional [15], [16], [17] neural networks to learn and perform compositions over word embeddings.

In sentence modeling, each sentence is regarded as a sequence of words. However, the Bag-of-Words (BoW) and topic models ignore word order information in the sentence. Mutual interaction of words in a sentence driven by composition models especially deep models can overcome the above shortcoming. In addition, a classifier can be incorporated into the architecture as the top layer to fine-tune the whole structure and word embedding for a specific task. In recent years, deep composition models are often reported to produce state-of-art results in various natural language processing tasks [18], [12], [15], [17], [16].

Whatever compositional model is used, the composition process is performed on the embeddings of each word in the sentence. In most scenarios, these word embeddings are pre-trained word vectors, which can be regarded as general representation since these embeddings are learned from a large corpus (over billions words). When applied to a specific application, these word embeddings can be fine-tuned to improve the performance. Despite the remarkable performance of the compositional models on various sentence modeling tasks, these models ignore the multi-sense nature of words. bThe majority of deep compositional models take single prototype word embeddings as input, in which each word is associated with a single embedding. However, a word may have multiple senses (for example, bank has two totally different meanings: sloping land and financial institution), and the appropriate sense of a word should be determined by the context of its occurrence. To address these problems, some works have proposed multi-sense embeddings-based models that represent each word by multiple sense-specific embeddings [19], [20], [21], [22], [23], and a few pioneering work have attempted to integrate word sense disambiguation into various compositional models for semantic capture [24], [25], [26], [27], [28]. In these multi-sense word embeddings based compositional semantic models, two operations are performed before the semantic composition: one is to learn multi-sense embeddings and another is to disambiguate the word sense. These two parts are complex and challenging, and are usually conducted separately.

R. Zhao and K. Mao are with with the School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798.
E-mail: rzhao001,ekzmao@ntu.edu.sg

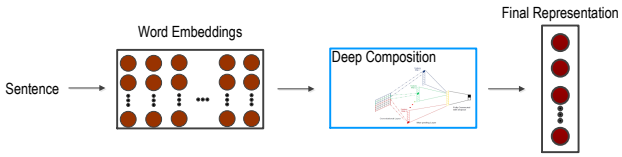

Fig. 1. Illustration of Deep Compositional Models built on top of Word Embeddings. In this framework, word embeddings are pre-trained over a large corpus, and are then fine-tuned in a task-specific corpus. The final representation is usually fed into a classifier, and the prediction error is back-propagated to tune model parameters.

In this paper, to address the aforementioned problems, we propose a lightweight and flexible framework for disambiguation in compositional models by introducing topic models. This framework aims to combine the two steps together. Here, we focus on two specific forms of composition, i.e., Convolutional Neural Network (CNN)[29], and Long Short Term Memory Network (LSTM)[30]. In recent years, some CNN-based and LSTMs-based sentence modeling works have been reported [15], [16], [31], [32], [33]. However, these models do not consider the multi-sense issue. Relying on the widely used topic model: Latent Dirichlet allocation (LDA) [34], the word-topic distribution $p(w|z)$ and document-topic distribution $p(z|d)$ can be inferred from the co-occurrence of words within the task-specific corpus. A topic z_i can thus be allocated to each word w_j in a specific document d_h according to the probability $p(z_i|w_j, d_h) \propto p(w_j|z_i)p(z_i|d_h)$. Since the length of a sentence is limited, and the entire sentence is an appropriate context for word sense disambiguation, the probability $p(z_i|w_j, d_h)$ can therefore be used to induce different sense for the same word under different context. Here, the context is defined as the sentence itself. The probability information along with topics embeddings encoded by fixed-length vectors can be integrated at two different levels of CNN and LSTMs, including word-level and sentence-level. Due to page limits, details of Topic-Aware CNN (TopCNN) will be elaborated and Topic-Aware LSTMs (TopLSTMs) will be briefly explained.

This paper is organized as follows. After reviewing related work in Section II, we introduce our proposed sentence representation learning models: Topic-Aware Convolutional Neural Network (TopCNN) and Topic-Aware Long Short Term Memory Networks (TopLSTMs) in Section III. In Section IV, experimental results on several sentence classification tasks are presented and analyzed. Finally, concluding remarks are provided in Section V.

II. Related Work

Since the present study is based on topic models and semantic compositional models, some valuable works in these research areas are briefly reviewed. Some works on multi-sense word embeddings for semantic composition are also reviewed.

A. Topic Model

Topic modeling approaches, such as Latent Dirichlet Allocation (LDA) and Probability Latent Semantic Analysis (PLSA), are derived from Latent Semantic Analysis, with the goal of reducing high dimensions of the document-term occurrence matrix into low dimensions denoted as latent topics [35], [36], [34]. The basic idea behind these methods is that the word choice in a document is influenced by the topic of the document probabilistically. The generation process of each word in a document is considered as a combination of three steps: a distribution over topics for the document is first selected randomly, a topic is then chosen from the above distribution, and finally a word is picked from the corresponding distribution over the vocabulary.

In the present work, LDA is employed to obtain word-topic and topic-document probabilities to enrich word embeddings for the subsequent use in the deep compositional model. Different from PLSA, LDA employs Dirichlet distribution as the prior probability for topic modeling.

B. Semantic Compositional Models

Semantic compositional models aim to represent larger natural language units such as phrases, sentences and documents in a vector space [37]. The compositional operation is conducted at the word-level, in which the meaning of words are also represented by vectors according to distributional information theory [38]. According to the nature of the employed composition, the compositional models can be categorized into shallow and deep models as described below.

Shallow Compositional Models: The interaction between words is modeled by algebraic operations including vector addition, vector multiplication, and tensors operations [10], [39], [40], [41]. In the above works, the algebraic operations are fixed. In other works, the algebraic operations can be learned based on some supervised settings defined by syntactic relations or semantic types [42], [43], [24].

Studies have found that shallow compositional models have limitations. For example, the addition model ignores the ordering of words and hence may not capture the true semantic interactions among words.

Deep Compositional Models: With the development of deep learning, recent works have focused on neural networks of various forms [18], [12], [15], [17], [16] with a strong and robust representation learning capability. The essence of deep compositional models is the application of multi-layers neural network of various forms including recursive [11], [12], recurrent [13], [14] and convolutional [15], [16], [17] neural networks to learn and perform compositions over word embeddings. In more recent works, structurally more complicated neural network models including gated recursive neural network [32], directed acyclic graphs LSTM [44] and dependency-sensitivity CNN built on LSTM [45] are employed. In sentence modeling, sentence is regarded as a sequence of words, in which each word is represented by the corresponding embeddings. Different neural networks employ different composition process. For example, recurrent neural networks perform composition along the word sequence one by one, while recursive neural networks perform composition along a pre-defined binary parsing tree, and convolutional neural networks (CNN) perform composition along sliding

windows. When the composition process passes through the entire sequence, the embeddings of all words in the sentence are fused into a final and fixed-length vector to represent the sentence. Since deep compositional models are more capable of capturing the semantic information underlying natural language units (especially higher units including sentences or documents), our work focuses on deep compositional models.

As discussed in the Introduction, the adoption of *single-sense word embeddings* hinders accurate semantic modeling no matter what neural networks are used unless multi-sense word embeddings are adopted. However, multi-sense word embeddings have their own problems which will be discussed next in section II-C. To alleviate these limitations, we integrate topic models into two state-of-arts deep compositional models: CNN and LSTMs and propose Topic-Aware CNN (TopCNN) and Topic-Aware LSTM (TopLSTM). The topic information inferred from topic models is used to enrich the word-embeddings before composition and the learned sentence-embeddings after composition in the framework of TopCNN and TopLSTMs.

C. Multi-sense Word Embeddings for Semantic Composition

In most word embeddings models, one word is represented by a unique vector without considering the polysemy and homonymy in natural language [7], [9]. The learned embeddings may be the average sense of its multiple senses. To alleviate the problem, some recent works have proposed multi-sense embeddings [19], [20], [21], [23], [46], [22]. Pre-clustering of the context of each word is often used to determine its specific sense [19], [20]. In [23], topical word embeddings are proposed, in which word embeddings are learned based on words occurrence and their topic information learned from LDA. Some other works rely on external resources, such as Wikipedia and WordNet for word disambiguation [21], [46]. In [22], a threshold value based on a non-parametric model is used to determine the number of senses of each word adaptively based on context pre-clustering. Most of the above works focus on the semantic modeling at word-levels without considering the semantic composition procedure. It should be noted that the topical word embeddings in [23] also employs LDA for word disambiguation. Different from the above topical word embeddings work that mainly focuses on word-level representation, our work here aims to develop a semantic composition model for sentence representation that considers multi-sense nature of words.

In a recent introductory and exploration work of multi-sense word embeddings based semantic composition [25], the pipelined framework shown in Fig. 2 incorporates word disambiguation into semantic composition via the following three steps: 1) *multi-sense embeddings learning*: each word is represented by multiple sense-specific embeddings; 2) *sense induction*: for a target language unit (sentence), the sense of each word and the corresponding embeddings are determined; and 3) *representation learning based on sense-specific embeddings*: compositional models are applied on the sense-specific embeddings. Shallow models including additive, multiplication and tensor operations combined with prior disambiguation are proposed in [27], [24], while deep compositional models based on neural networks are adopted in [25], [28], [26].

Our proposed topic-aware deep compositional framework is different from the above mentioned works in the following aspects. Firstly, our framework is a lightweight and flexible without requiring *multi-sense embeddings learning* and *sense induction* as separated steps to the final compositional module. Via topic awareness, the embeddings of words used in our model consists of two parts: the general sense learned from a large corpus and the topic-specific sense learned from the task-specific corpus. Therefore, our framework does not require additional multi-sense embeddings models. Secondly, in our proposed framework, the applied LDA model utilizes the whole co-occurrence statistics of the corpus for word disambiguation instead of the neighbourhood contexts required by previous works [23], [47]. In these previous works, the hyper-selection of lengths about context is a very tricky issue. Thirdly, our framework is more extensible than these previous works, and various deep compositional models with different neural networks can be developed under this framework. In our work, two state-of-arts compositional models including CNN and LSTMs are adopted as examples to derive two topic-aware compositional models, namely TopCNN and TopLSTMs, respectively.

In the literature, there are some works that incorporate multiple versions of word embeddings into CNN model for sentence classification, where one version of word embeddings can be regarded as one channel [48], [49]. Our proposed TopCNN models utilize word embeddings and topic embeddings, which can also be regarded as two sets of word embeddings. However, the previous works still can not address the multi-sense issues, since word embeddings in each channel are shared by all occurrences of one certain word in corpus. By contrast, topic embeddings in our work are determined by occurrences of words in different context, i.e., the probability information of words. This means that the topic embeddings of word is derived by its context, and is able to derive multi-sense word embeddings. In addition, the topic embeddings in our work can be incorporated into word and sentence levels, while the previous works can only integrate multi-prototype word embeddings at word level.

III. Topic-Aware Deep Compositional Models

Assuming that word embeddings are pre-trained from a large corpus denoted as a matrix $\mathbf{W} \in \mathbb{R}^{V \times d}$, where V is the vocabulary size and d is the dimension of word embeddings. LDA is applied to the task-specific corpus to derive the probability distributions of words and sentences over latent topics, i.e., $p(w_i|z_i)$ and $p(z_i|s)$, where z_i belongs to the pre-defined topic set $Z = \{z_1, z_2, \ldots, z_T\}$. Here, we adopted a variant of LDA: Biterm Topic Model [50], which is effective for short text modeling. Embeddings for topics, denoted by matrix $\mathbf{W}_t \in \mathbb{R}^{T \times d_t}$, will also be learned. For word embeddings, there are various public resources including pre-trained word embeddings and tools for implementation. For topic embeddings, we will present four simple but effective schemes to learn representations for each topic after introducing the

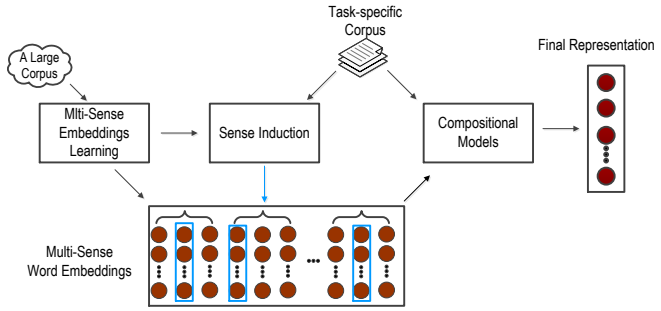

Fig. 2. Illustration of Semantic Composition Models built on top of Multi-Sense Embedding. In this framework, multi-sense embeddings model are first trained on a large-corpus. Then, based on the interested language unit in the task-specific corpus, the word sense is inferred and the corresponding sense-specific embeddings is picked up. Finally the compositional models are applied on the selected sense-specific embeddings to produce the final representation.

whole structures of our TopCNN and TopLSTMs. At last, the additional computational burden caused by LDA will be discussed.

A. Topic-Aware CNN

The structure of the proposed Topic-Aware Convolutional Neural Network (TopCNN) is shown in Figure 3. For a sentence $s = (w_1, w_2, \ldots, w_l)$ with l words, the TopCNN aims to learn its embeddings.

Word-Level : For word w_i in sentence s, its probability over the latent topics $p(z_i|w_j, s) \propto p(w_j|z_i)p(z_i|s)$ is calculated. After normalization, the topical distribution of word w_i is denoted by vector $\mathbf{p}_{w_i}$:

$$\begin{aligned} \mathbf{p}_{w_i} = [p(z_1|w_i, s), p(z_2|w_i, s), \ldots, p(z_T|w_i, s)] \\ \mathbf{s.t}\ p(z_i|w_i, s) \geq 0\ and\ |\mathbf{p}_{w_i}| = 1 \end{aligned} \tag{1}$$

where the i-th element in $\mathbf{p}_{w_i}$ denotes the probability that word w_i is assigned to the i-th topic. This topical distribution $\mathbf{p}_{w_i}$ is a useful resource for word disambiguation, which represents each word occurred in a specific context by the corresponding topic distribution. The topical information is derived by applying LDA to the word-document occurrence statistics instead of the word and its window-based context. To utilize the topical information of each word in the compositional model, the topic-specific embeddings for each word w_i based on the topical information $\mathbf{p}_{w_i}$ and the embeddings for topics $\mathbf{W}_t$ is derived as follows:

$$\mathbf{t}_{w_i} = \sum_{i=1}^{T} p(z_i|w_i, s) * \mathbf{W}_t[i] = \mathbf{p}_{w_i} \times \mathbf{W}_t \tag{2}$$

The general word embeddings pre-trained over a large corpus is concatenated with the topic-specific embeddings to form the representation of the word as $\mathbf{x_i} \in \mathbb{R}^{d+d_t}$:

$$\mathbf{x}_i = \mathbf{W}[w_i] \oplus \mathbf{t}_{w_i} = [\mathbf{W}[w_i]; \mathbf{t}_{w_i}] \tag{3}$$

where $\oplus$ denotes concatenation operation. As shown in Eq. 3, the final representation of a word consists of two parts: one is the general word embeddings and another is the topic-specific embeddings. General word embeddings provide the general sense of the word, while the topic-specific embeddings are able to disambiguate the senses of the words.

Convolution: the convolution operation is defined as a multiplication operation between a filter vector $\mathbf{u} \in \mathbb{R}^{m(d+d_t)}$ and an concatenation vector representation $\mathbf{x}_{i:i+m-1}$ given by:

$$\mathbf{x}_{i:i+m-1} = \mathbf{x}_i \oplus \mathbf{x}_{i+1} \oplus \cdots \oplus \mathbf{x}_{i+m-1} \tag{4}$$

where $\mathbf{x}_{i:i+m-1}$ represents a window of m words starting from the word indexed as i. A bias term b is added to the output of the convolution operation, and thus the final operation is given as:

$$c_i = g(\mathbf{u}^T \mathbf{x}_{i:i+m-1} + b) \tag{5}$$

where $*^T$ denotes the transpose of a matrix $*$ and g is a non-linear activation function such as Sigmoid, Hyperbolic Tangent, or ReLu. In our work, ReLU is adopted due to its reported good performance [51], but other activation functions are equally applicable.

Each vector $\mathbf{u}$ can be regarded as a filter. The scalar value c_i can be regarded as the learned feature in the i-th window of words under filter $\mathbf{u}$. The width of the filter, i.e., window size m, can be varied. The setting of window size will be discussed in the experiment section IV-B.

Max-pooling: The convolution operation over the whole sequence of words is applied by sliding the window from the beginning to the ending of the sentence. Since each sentence of length l has multi-windows of words $\{\mathbf{x}_{1:m}, \mathbf{x}_{2:m+1}, \ldots, \mathbf{x}_{l-m+1:l}\}$, a feature map can be obtained and the dimension of the feature map is determined by the length of the sentence l and the window size m:

$$\mathbf{c}_j = [c_1, c_2, \ldots, c_{l-m+1}] \tag{6}$$

where the index j denotes the j-th filter. By conducting the max-pooling operation in the feature map obtained in the convolutional layer, the single feature corresponding to the filter is obtained as:

$$h_j = max(\mathbf{c}_j) \tag{7}$$

where h_j corresponds to the j-th dimension of the output of the pooling layer. Generally, multiple filters are applied with different weights and different window sizes to derive a feature vector. The motivation behind the max-pooling operation is to filter out meaningless combinations of words. Another effect of max-pooling is that it produces a fixed-length feature vector regardless the sentence lengths.

Sentence-Level: Considering various semantic and syntactic patterns in natural language, we design multiple filters with different window sizes and randomly initialized weights to match and capture these patterns. After convolution and max-pooling operations, the produced scalar is the output of a filter applied on the entire sentence. Assuming there are k filters, the output of the pooling layer is the concatenation of all outputs of the filters:

$$\mathbf{h} = [h_1, h_2, \ldots, h_k] \tag{8}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2632521, IEEE/ACM Transactions on Audio, Speech, and Language Processing

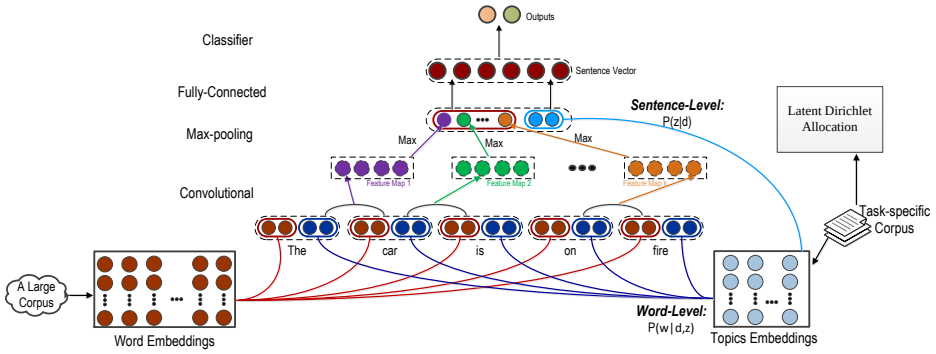

Fig. 3. The Framework of our TopCNN models for sentence representation learning. It illustrates the whole process to learn the sentence vector from the input sentence. Word Embeddings are learned in a large corpus, which is regarded as the general sense of words. Then, with the favor of LDA trained in the task-specific corpus, the probability information can be utilized at word-level and sentence-level, respectively. At the word-level, the topic-specific embeddings of words are concatenated behind the general word embeddings. At the sentence-level, the topic-specific embeddings of words are concatenated behind the output vector of max-pooling layer. After multiple fully-connected layers, the learned sentence representations are finally fed into the classifier.

$\mathbf{h}$ can be used as the final representation of the input sentence, if the different filters are enough to discover all meaningful structure patterns of the sentence. However, $\mathbf{h}$ may not convey the complete semantic meaning of the entire sentence, since each element in $\mathbf{h}$ is only the activation of a local window of words. Although the window size of applied filters can be set to the length of the sentence, the high computational cost and the sparsity problem are the inherent challenges. To address the problem, we propose to enrich the representation based on topical information similar to the integration of topic model at word-level. After applying LDA, the probabilities of the sentence over each topic is given by:

$$\begin{aligned} \mathbf{p}_s &= [p(z_1|s), p(z_2|s), \ldots, p(z_T|s)] \\ &\mathbf{s.t}\ p(z_i|s) \geq 0\ and\ |\mathbf{p}_s| = 1 \end{aligned} \tag{9}$$

The sentence's probabilities are then transformed into an embedding based on embeddings for topics, which is given by:

$$\mathbf{t}_s = \sum_{i=1}^{T} p(z_i|s) * \mathbf{W}_t[i] = \mathbf{p}_s \times \mathbf{W}_t \tag{10}$$

where $\mathbf{t}_s \in \mathbb{R}^{d_t}$ is the topic-specific embedding for the sentence. Different from $\mathbf{h}$, which is composed of the activations of the local windows of words over the sentence, $\mathbf{t}_s$ is the activation of the global context of the sentence. Therefore, the two embeddings can be concatenated to represent the sentence as:

$$\mathbf{s} = \mathbf{h} \oplus \mathbf{t}_s = [\mathbf{h}; \mathbf{t}_s] \tag{11}$$

Fully-Connected Layers and Classifier: The learned representation of the sentence is passed to multiple hidden layers to seek a higher-level sentence representation. Several fully-connected dense layers are stacked together, in which the output of one layer is used as the input of the next layer. The computation in each layer is given by:

$$\mathbf{o}_i = g(\mathbf{B}_i \mathbf{q}_i + \mathbf{b}_i) \tag{12}$$

where $\mathbf{o}_i$ and $\mathbf{q}_i$ denote the output and input of the i-th fully-connected layer, respectively. $\mathbf{B}_i$ and $\mathbf{b}_i$ represent the transformation matrix and the bias term in the i-th fully-connected layer, respectively. The function g is also set to the rectified linear unit (ReLU). It is clear that the input of the first fully-connected layer is the initial representation of sentence, i.e., $\mathbf{q}_0 = \mathbf{s}$. The output of the last layer: $\mathbf{o}_{r-1}$ is regarded as the final representation of the input sentence, assuming r fully-connected dense layers are successively stacked.

The final learned representation of the input sentence is fed into the classification layer. Here, we adopt a fully connected softmax layer to predict the probability distributions of the sentence over the whole label set.

Training and Regularization: Given the predicted outputs and true labels, the cross-entropy errors over training corpus can be calculated and back-propagated to update model parameters. The model parameters include filter weights, parameters of fully-connected hidden layers and softmax classification layer. In addition, word embeddings $\mathbf{W}$ and topic embeddings $\mathbf{W}_t$ are also fine-tuned. To avoid overfitting, regularization technique is applied. Here, we follow some settings of the CNN structure proposed in [15]. Firstly, dropout noise with masking probability p is applied on the feature vector fed into the fully-connected softmax layer [52]. Then, a $l2-$norm constraint over the filter weights $\mathbf{u}$ and fully-connected hidden layers' transformation matrix $\mathbf{B}_i$ is imposed during training. The whole network is trained by stochastic gradient descent with Adadelta update rule that has been shown as an effective and efficient backpropagation algorithm [53]. The training corpus are shuffled randomly to generate a stream of mini-batches for model training.

B. Topic-Aware LSTM

The structure of our proposed Topic-Aware Long Short Term Memory Networks (TopLSTMs) is shown in Figure 4. Details of LSTM can be found in [30]. Similar to TopCNN, topic embeddings of words are incorporated into the corresponding word embeddings, and the concatenation of these general and topic-specific word embeddings is fed into LSTM

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2632521, IEEE/ACM Transactions on Audio, Speech, and Language Processing

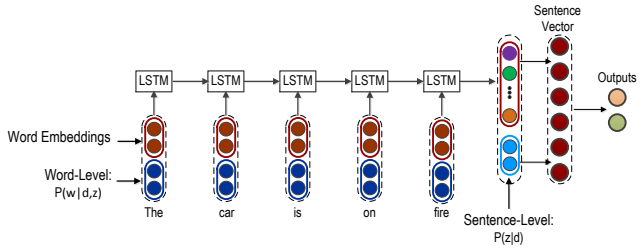

Fig. 4. The Framework of our TopLSTMs for sentence representation learning.

units. The topic embeddings of sentence can also be concatenated with the outputs of the proceeding step. The learned representation for the sentence is finally fed into multiple hidden layers and the final classification layer.

C. Embeddings Learning for Topics

In the above sections, the probability information derived by LDA in the form of embeddings for topics $\mathbf{W}_t$ is incorporated into the convolutional semantic composition process. In the following, we present four models of topic embeddings learning including random, additive, topical neural models and joint neural models.

Random Model: These topic embeddings are first randomly initialized, and are then fine-tuned during the training stage.

Additive Model: The motivation behind the additive model is very intuitive that the topic can be represented by a list of top words ranked by probabilities belonging to this topic. Therefore, the meaning of the topic can be approximated by the meanings of these top words $\mathbf{W}_t$. We average the embeddings of these top-ranked words to obtain the embeddings of the topic:

$$\mathbf{W}_t\left[i\right] = \frac{1}{h}\sum_{j=0}^{h-1}\mathbf{W}[w_j] \tag{13}$$

where $\mathbf{W}_t$ denote the embeddings for the t-th topic, w_j is one of the top h words whose probabilities belonging to topic are the largest, and $\mathbf{W}$ is the pre-trained word embeddings matrix. It is clear that under the additive model, the embeddings for topics and words have the same dimensionality.

Topical Neural Model: The neural model is inspired by the efficient word embedding framework word2vec, including Continuous Bag-of-Word (CBoW) model and Skip-Gram [9]. The training objective of CBoW is to predict the target word by combining the embeddings of context words, while the objective of Skip-Gram is to predict the context word by the target and central word in the context. In topic embedding learning, the word occurred in documents can be replaced by the corresponding topics. Based on LDA, we firstly assign the topic to each word w_i based on Eq. 1, from which the topic having the highest probability is picked up. The assigned topic is then regarded as a pseudo word to replace the actual word. Thus, the idea of word embedding learning can naturally be used to learn topic embeddings. Similar to word2vec, the maximization objectives of our neural model-based topic embedding learning are defined as follows:

$$L(s)^{TNM}_{cbow} = \frac{1}{M}\sum_{i=1}^{M}\log(\mathbf{z}_i|\mathbf{z}_{context}) \tag{14}$$

$$L(s)^{TNM}_{skip} = \frac{1}{M}\sum_{i=1}^{M}\sum_{-k\le j\le k}\log(\mathbf{z}_{i+j}|\mathbf{z}_i) \tag{15}$$

where $\mathbf{z}_i$ denotes the embedding for the topic assigned to the i-th word in the sentence s. In Figure 5(a), the architecture of topical neural model is shown.

Joint Neural Model: One concern behind the above topical neural model lies in the possible large divergence between the learned topic embeddings and the word embeddings. Since topical neural model regards each topic as a pseudo word, the topic embeddings are learned without word information. To address this issue, a joint neural model is proposed by taking each word-topic pair instead of topic as a pseudo word. Similar to Eqs. 14 and 15, the target functions of joint neural model are defined as:

$$L(s)^{JNM}_{bow} = \frac{1}{M}\sum_{i=1}^{M}\log\left(\{\mathbf{w}_i,\mathbf{z}_i\}\Big|[\mathbf{w}_{context},\mathbf{z}_{context}]\right) \tag{16}$$

$$L(s)^{JNM}_{skip} = \frac{1}{M}\sum_{i=1}^{M}\sum_{-k\le j\le k}\log\left(\{\mathbf{w}_{i+j},\mathbf{z}_{i+j}\}\Big|\{\mathbf{w}_i,\mathbf{z}_i\}\right) \tag{17}$$

For each word topic pair $\{\mathbf{w}_i,\mathbf{z}_i\}$, the word embeddings and topic embeddings are concatenated together. Compared to the topical neural model, the joint neural model may encounter sparsity problem. However, in this model, the word embeddings are pre-trained and fixed during the training, while the topic embeddings are fine-tuned in the local corpus. The advantages of this settings are two-folders: one is the alleviation of the sparsity issue during model training and another is the reduction of the domain difference between learned word embeddings and topic embeddings. It should be noted that the proposed Joint Neural Model is similar to the TWE-3 model proposed in [23]. However, our model uses word2vec embeddings trained on a corpus with billions of words, while TWE-3 learns word and topics embeddings from task-specific corpus. The joint neural model is illustrated in Figure 5(b).

Among the four topic embeddings learning methods including random, additive, topical neural and joint neural models, the first two methods belong to lazy methods without any learning schemes, while the last two are based on the word2vec model. The difference between the topical neural model and the joint neural model is that the joint neural model considers word and topic information jointly in the training process, while the topical neural model relies on topic information only. A detailed experimental analysis over the four methods is given in section IV-E.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2632521, IEEE/ACM Transactions on Audio, Speech, and Language Processing

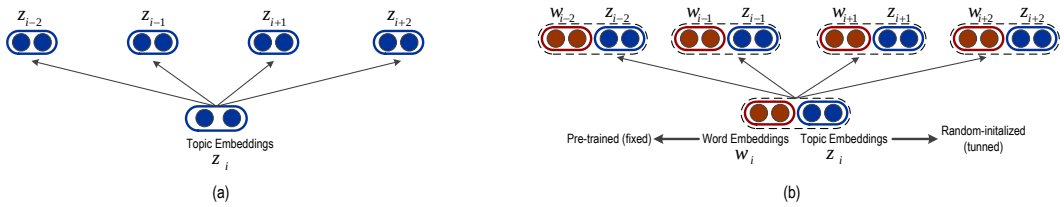

Fig. 5. Two neural network architectures (Skip-gram based) for topic embeddings learning: (a) Topical Neural Model; (b) Joint Neural Model.

D. Complexity Analysis

The proposed Topic-Aware deep compositional framework is a pipelined system built upon LDA and deep compositional models. Compared to the basic deep composition models CNN or LSTMs, the complexity is increased due to the use of LDA. In this study, we use a LDA variant named biterm topic model (BTM) to model topics in short texts [50]. The computational complexity of the Gibbs sampling implementation of BTM, i.e., the additional computational burden of TopCNN, is $\mathcal{O}(KD\frac{\bar{l}(\bar{l}-1)}{2})$, where K, D and $\bar{l}$ denote the topic number, the document number and the average length of document, respectively. Since deep compositional models including CNN and LSTMs usually have huge numbers of parameters to learn, the increased computational complexity due to topic models is negligible.

IV. Experiments

In this section, we evaluate our proposed TopCNN and TopLSTMs on sentence classification tasks. Five benchmark real datasets are used. Several state-of-arts methods including traditional classification methods and sentence models are compared and the sensitivity of our models to the number of topics in LDA is analyzed. Finally, the topic embedding learning methods and their effects on the performance of our proposed topic-aware deep compositional models are discussed.

A. Experimental Settings

Datasets: Five benchmark datasets are used in the experiment. The datasets and their tasks are described below:

- **MR**: Movie Review dataset [2][1]. Each sample is a movie review sentence. The binary sentiment polarity is defined as positive or negative. The task is to classify each review by its binary sentiment polarity.
- **Subj**: Subjectivity dataset [54][2]. Each sample is a sentence. The task is to classify each snippet into subjective or objective categories.
- **CR**: Customer Reviews dataset [55][3]. The corpus consists of customer reviews of several products from Amazon. The task is to classify each customer review by its binary sentiment polarity: positive or negative.
- **MPQA**: Opinion on a phrase level polarity detection subtask of the MPQA dataset [56][4]. The task is to judge each phrase as positive or negative.
- **TREC**: Question classification dataset containing 6 different types of classification in [57][5]. The task is to classify the question content by their question type.

Statistics of these datasets are given in Table I. Since the first four datasets do not have explicit training/testing split, 10-folder cross validation (CV) is applied on them. TREC dataset provides the training/testing split, in which training data have 5452 instances and testing data have 500 instances.

Compared Approaches: Several state-of-arts text classification and sentence modeling methods are compared here. The details of the compared approaches are given below:

* **NBSVM** and **MNB**: Naive Bayes SVM and Multinomial Naive Bayes with unigram and bigram features [58].
* **G-Dropout** and **F-Dropout**: Gaussian Dropout and Fast Dropout [59]
* **RAE** and **MV-RNN**: RAE is Recursive Autoencoders on word embeddings trained from Wikipedia [11]. The composition path is defined by a binary parse tree. MV-RNN is Matrix-Vector Recursive Neural Network with a parse tree [18].
* **AdaSent**: AdaSent is a self-adaptive hierarchical sentence model [32]. Adasent adopts a hierarchy structure, where consecutive levels are connected through gated recursive composition of adjacent segments, and feeds the hierarchy as a multi-scale representation through a gating network to a particular learning task.
* **CNN**, **CNN$_{600}$**, **DCNN** and **DSCNN**: CNN is Convolutional Neural Network on pre-trained word embeddings from word2vec, Google News [15]. Word embeddings, whose dimensionality is 300, are also fine-tuned to learn task-specific embeddings and parallel CNN with the static word embeddings. CNN$_{600}$ is the same as CNN except that the dimensionality of adopted word embeddings is 600. DCNN is Dynamic Convlolutioanl Neural Network that adopts the dynamic k-max pooling instead of max-pooling [16]. DSCNN is dependency-sensitivity CNN, in which CNN is adopted on top of LSTM in a multi-channel [45].
* **P.V.**: P.V. is Paragraph Vector that learns distributed representations of words and short-text jointly [60]. We feed the learned paragraph vectors into linear SVM for

[1]https://www.cs.cornell.edu/people/pabo/movie-review-data/
[2]https://www.cs.cornell.edu/people/pabo/movie-review-data/
[3]http://www.cs.uic.edu/iub/FBS/sentiment-analysis.html
[4]http://mpqa.cs.pitt.edu/
[5]http://cogcomp.cs.illinois.edu/Data/QA/QC/

TABLE I
STATISTICS FOR THE FIVE USED DATASETS IN OUR PAPER. N AND v COUNT THE NUMBER OF INSTANCES AND THE VOCABULARY SIZE, RESPECTIVELY. $\bar{l}$ DENOTES THE AVERAGE LENGTH OF EACH DATASET. WE USE 10-FOLD CROSS-VALIDATION FOR THESE DATASETS WITHOUT EXPLICIT SPLITTING INCLUDING MR, SUBJ, CR AND MPQA. AND FOR EACH DATASET, WE ALSO PROVIDE THE NUMBER OF CLASSES DENOTED AS $|\mathbb{C}|$.

Dataset	N	v	$\bar{l}$	$\|\mathbb{C}\|$	train/test
MR	10662	18765	20	2	CV
Subj	10000	21323	23	2	CV
CR	3775	5340	19	2	CV
MPQA	10606	6246	3	2	CV
TREC	5952	9592	14	6	5452/500

classification and the implementation of P.V. is based on *gensim*[6].

* **RNN**, **LSTM** and **GRUs** all belong to recurrent models [61], [62], [30]. Here, we use the library *keras* to implement these methods. The hidden layer sizes are all set to be same as the embedding size. The same word embeddings adopted by CNN are used for these three recurrent models.

We implemented CNN, CNN_{600}, RNN, P.V, LSTM and GRUs models based on their public sources or packages. For word embeddings in these models, the public word vectors trained on part of Google News dataset by word2vec were adopted here[7]. The dimension of word embeddings is 300. The embeddings of the out-of-vocabulary words were randomly initialized. During model training, these pre-trained word embeddings are further fine-tuned for a fair comparison. For the rest of the models, their published results in their original papers are cited directly.

B. Specifications for TopCNN

Firstly, four variants of our proposed TopCNN are considered.

- o **$\text{TopCNN}_{\text{word}}$**: Only the word-topic probability information is used to enrich the word embeddings.
- o **$\text{TopCNN}_{\text{sen}}$**: Only the sentence-topic probability information is used to enrich the representation output of the pooling layer.
- o **$\text{TopCNN}_{\text{word\&sen}}$**: Both word-topic and sentence-topic probability information are used.
- o **$\text{TopCNN}_{\text{ens}}$**: An ensemble model of the above three variants of TopCNN models by averaging the class probability scores generated by the above three models together.

We used the same pre-trained Google word embeddings in our models and the compared models. For topic model, we used Biterm Topic Model[8] with a topic number of 100, i.e., $T = 100$. For embeddings of topics, we used the joint neural model, in which the window size is 2, and the topic embeddings have the same dimension as word embeddings, i.e., $d_t = d = 300$. For the convolutional layer, a broad range of filter windows sizes (2,3,4,5,6,7) were used, in which each window size corresponds to 200 feature maps. Therefore, we totally have $k = 1200$ filters. For the fully-connected hidden layers, the number of layers are set to 2, and the output dimensions of these two layers are set to 1200 and 300, respectively. The other parameters including L2-norm constraint over model parameters, the dropout rate and the training batch size were all set using the validation datasets, which are held-out portions of the training data. Finally, the whole training data were used to train the models with these determined hyperparameters.

Similarly, four variants of our proposed TopLSTMs are considered.

- o **$\text{TopLSTMs}_{\text{word}}$**: Only the word-topic probability information is used to enrich the word embeddings.
- o **$\text{TopLSTMs}_{\text{sen}}$**: Only the sentence-topic probability information is used to enrich the representation output of the pooling layer.
- o **$\text{TopLSTMs}_{\text{word\&sen}}$**: Both word-topic and sentence-topic probability information are used.
- o **$\text{TopLSTMs}_{\text{ens}}$**: An ensemble model of the above three variants of TopLSTMs models by averaging the class probability scores generated by the three variants together.

C. Experimental Results

The results are shown in Table II. We compare our proposed methods with several state-of-arts methods including text classification models, CNN-based and RNN-based compositional models. The first observation is that our ensemble model $\text{TopCNN}_{\text{ens}}$, achieves comparable performances with the AdaSent model which produces up-to-date benchmark results on the five datasets[9]. It can be observed that $\text{TopCNN}_{\text{ens}}$ outperforms AdaSent in two datasets including *CR* and *TREC*. Compared with traditional text classification models including NBSVM, MNB, G-Dropout and F-Dropout, our proposed TopCNN and TopLSTMs models are able to offer better performance without requiring feature engineering on all datasets. This means that the deep compositional model can indeed boost the representation learning for sentences. As shown in Table II, P.V. model is not able to achieve satisfying performance over five datasets. This is probably because P.V. is not good at modeling embeddings for short documents due to limited context information.

In experiments, we also compared our methods with other deep compositional models including CNN-based and RNN-based compositional models. Our proposed TopCNN models, especially $\text{TopCNN}_{\text{word\&sen}}$ and $\text{TopCNN}_{\text{ens}}$, consistently outperform these models on five datasets. Compared to the original CNN model, the achieved performance improvements verify the effectiveness of topical embeddings. In addition,

[6]https://radimrehurek.com/gensim/models/doc2vec.html

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

[8]The code for BTM has been kindly provided at https://github.com/xiaohuiyan/BTM

[9]DTCNN gives the best performance as 95.6. However, it relies on the quality of the parsing operation. As reported in [31], the adopted parser is trained on the annotated *TREC* dataset so that a high-quality of parsing structure can be guaranteed in this corpus.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2632521, IEEE/ACM Transactions on Audio, Speech, and Language Processing

CNN_{600} performs worse than CNN on these five datasets, which means the performance improvement of our proposed TopCNN is not resulted from the increased dimensionality of word embeddings. Although the TopLSTMs models do not consistently outperform CNN models, they outperform their basic model: LSTM model. The performance gains achieved by $TopCNN_{ens}$ and $TopLSTMs_{ens}$ over the basic CNN and LSTM models on the five datasets are shown in Figure 6. The significant performance gains of our models show that the integration of LDA into the deep compositional process can lead to a performance boost. The composition process in our proposed models combine not only general sense of words, but also topic-specific sense of words. In addition, at the sentence level, the output of local receptive fields can be combined with the activation of the whole scope of sentences provided by topic-document probability information. The ensemble of three different levels of topic awareness further improves the model performance.

As shown in Table II, the *ensemble* version of Topic-aware model performs better than other variants. Ensemble methods always outperform the constituent learning algorithms. Experimental results also show that among the rest three variants of TopCNN and TopLSTMs models, the *word&sen-level* model perform best and the *sentence-level* model perform worst in all datasets. The reason behind the deteriorated performance of the *sentence-level* model may be that a direct combination of the embeddings learned by basic compositional models and the topical embeddings for sentences is inappropriate due to the discrepancy between the two semantic spaces. In other two variants: the *word-level* and *word&sen-level* models, the convolutional operation is conducted on the combination of general-sense and topic-specific embeddings. The filter weights are learned during training so that the possible semantic space discrepancy may be reduced or even eliminated at the output of the filter. Compared to the *word-level* model, our *word&sen-level* model further integrates the sentence-topic probability to enrich the output of max-pooling layer or the final time step and feed them into multiple fully-connected layers before the final classification. The further utilization of $topic_{sen}$ probability may explain the better performance of the *word&sen-level* model than the *word-level* model .

Model variances of several comparable models are further investigated, which can reduce the possible bias caused by the randomness of parameters initialization. We conducted testing of these models including P.V., RNN, LSTM, GRUs, CNN and our proposed TopCNN and TopLSTMs models ten times and reported their mean accuracies and corresponding standard deviation in Table III. The model variances of AdaSent is from [32]. As shown in Table III, our proposed $TopCNN_{word\&sen}$ and $TopCNN_{ens}$ achieve the best performance among these methods.

D. Sensitivity to the Number of Topics

The number of topics in LDA is a hyperparameter, which is vital to the performance of topic modeling. If the topic number is too small, the model would be too simple to capture the true topical distribution. On the other hand, a very large topic number may lead to the over-fitting problem due to the model's high complexity [63]. Since our proposed model integrates the topical probability into the semantic composition process, we next investigate the sensitivity of our models in response to the change of topic numbers based on the quality of learned representation on sentences.

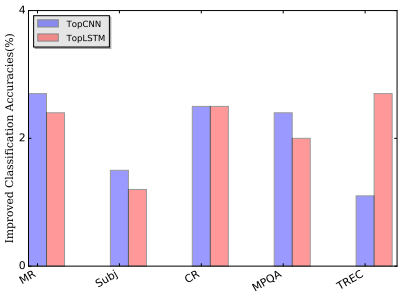

Fig. 6. Performances gains achieved by our proposed TopCNN and TopLSTMs models compared to CNN and LSTM, respectively

Here, we only compared the performance of our proposed three TopCNN models including $TopCNN_{word}$, $TopCNN_{sen}$ and $TopCNN_{word\&sen}$ under different numbers of topics over the five datasets. The topic number was set to 50, 100, 150, 200, 250, 300, respectively. We also compared the original CNN model without using topic models, which corresponds to zero topic number. Under different topic number, we run LDA and our proposed joint neural model to learn topic embeddings. Therefore, for each dataset and each model, we had 7 different topic number settings and the other model parameters were fixed as the above reported setting and the other experimental settings are kept unchanged as reported above. From the results shown in Figure 7, we have the following observations:

Firstly, TopCNN models with a moderate topic number achieve the best performance. This means that the integration of LDA into the semantic composition process is helpful for representation learning of sentence, and our proposed TopCNN models outperform the original CNN model under an appropriate topic number. It is noted that under large topic numbers, our proposed TopCNN models perform worse than the original CNN model. This may be because LDA may not be able to infer topics behind short text under a large topic number. In addition, the increased dimensionality of word embeddings hinders the performance of CNN models as reflected in the performance comparison between CNN and CNN_{600}.

E. Discussion on Topic Embeddings Learning

The experimental results have shown that the joint neural model presented in Section III-C can lead to a good topic embeddings learning for our proposed models. In the joint neural model, the occurrence of the pair of word and topic is utilized via word2vec framework to learn topic embeddings. Therefore, semantic space of topics embeddings can be close to that of word embeddings. In Section III-C, four topic embeddings learning schemes are proposed. Here, the four schemes incorporated in the proposed TopCNN are investigated experimentally.

For the random model, the topic embeddings were generated randomly following a uniform distribution in the range of $[-0.25, 0.25]$. For the additive model, we average the embeddings of top 10 words for each topic as described in Eq. (13). For the topical neural model, based on the target function Eq .(15), the topic embeddings with a dimensionality of 300 were trained on the five datasets. Finally, the performance of our TopCNN model with four kinds of topics embeddings was compared. Here, we also randomly selected one folder for the first four datasets and for each settings, we run the same model ten times and recorded its corresponding mean value and standard deviation.

The results are shown in Figure 8. Firstly, the random topic embeddings perform the worst, while the performance difference is not significant. This may be due to the adopted fine-tuning on topic embeddings. Figure 8 also shows that the topic embeddings learned from neural models do not consistently outperform over

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2016.2632521, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE II
CLASSIFICATION ACCURACIES (%) FOR COMPARED METHODS ON THE WHOLE FIVE ADOPTED DATASETS. BOLD FACE INDICATES HIGHEST ACCURACIES

Category	Method	Datasets				
		MR	Subj	CR	MPQA	TREC
Text Classification Models	NBSVM	79.4	93.2	81.8	86.3	-
	MNB	79.0	93.6	80.0	86.3	-
	G-Dropout	79.0	93.4	82.1	86.1	-
	F-Dropout	79.1	93.6	81.9	86.3	-
CNN and its Variants	CNN	81.3	93.5	83.9	89.4	93.0
	CNN_{600}	79.3	92.0	81.6	87.5	91.9
	DCNN	-	-	-	-	93.0
	DSCNN	82.2	93.2	-	-	**95.6**
	P.V.	75.9	92.2	77.9	75.4	91.5
Other Deep Compositional Models	RAE	77.7	-	-	86.4	-
	MV-RNN	79.9	-	-	-	-
	RNN	77.2	90.9	71.8	88.6	83.8
	LSTM	79.5	93.3	80.4	88.8	89.4
	GRUs	80.5	93.5	82.1	89.0	91.8
	AdaSent	**83.1**	**95.5**	**86.3**	**93.3**	91.8
Our Models	$TopCNN_{word}$	81.7	93.4	84.9	89.9	92.5
	$TopCNN_{sen}$	81.3	93.4	84.8	90.3	92.0
	$TopCNN_{word\&sen}$	82.3	94.3	85.6	91.1	93.6
	$TopCNN_{ens}$	**83.0**	95.0	**86.4**	91.8	94.1
	$TopLSTMs_{word}$	81.2	94.1	82.6	89.6	91.5
	$TopLSTMs_{sen}$	80.6	93.7	81.6	89.1	90.5
	$TopLSTMs_{word\&sen}$	80.8	94.0	82.3	89.5	91.4
	$TopLSTMs_{ens}$	81.9	94.5	82.9	90.8	91.9

TABLE III
MODEL VARIANCES FOR COMPARED METHODS BASED ON OUR OWN IMPLEMENTATIONS ON THE WHOLE FIVE ADOPTED DATASETS.

Methods	MR	Subj	CR	MPQA	TREC
P.V.	73.1 ± 0.6	90.5 ± 0.9	76.1 ± 0.4	72.2 ± 0.8	89.9 ± 0.9
RNN	77.2 ± 0.3	89.9 ± 0.5	71.9 ± 0.5	88.2 ± 0.1	83.6 ± 0.7
LSTM	78.9 ± 0.7	93.1 ± 0.6	80.1 ± 0.4	88.4 ± 0.1	89.1 ± 0.5
GRUs	80.6 ± 0.6	93.3 ± 0.7	82.4 ± 0.2	89.1 ± 0.5	92.3 ± 0.4
CNN	81.4 ± 0.3	93.3 ± 0.2	83.5 ± 0.3	89.3 ± 0.1	92.5 ± 0.5
AdaSent	79.8 ± 1.2	92.2 ± 1.2	83.6 ± 1.6	90.4 ± 0.7	91.1 ± 1.0
$TopCNN_{word}$	81.3 ± 0.3	93.5 ± 0.2	84.9 ± 0.2	89.9 ± 0.2	92.4 ± 0.6
$TopCNN_{sen}$	81.1 ± 0.4	93.1 ± 0.3	84.5 ± 0.3	90.1 ± 0.2	92.1 ± 0.2
$TopCNN_{word\&sen}$	82.4 ± 0.1	94.1 ± 0.2	85.3 ± 0.2	90.9 ± 0.1	93.5 ± 0.3
$TopCNN_{ens}$	82.8 ± 0.1	94.7 ± 0.1	86.4 ± 0.1	91.5 ± 0.1	93.7 ± 0.2
$TopLSTMs_{word}$	81.1 ± 0.5	93.8 ± 0.6	82.7 ± 0.3	89.9 ± 0.2	91.1 ± 0.5
$TopLSTMs_{sen}$	80.4 ± 1.1	93.5 ± 0.9	81.7 ± 0.5	89.2 ± 0.3	90.3 ± 0.8
$TopLSTMs_{word\&sen}$	80.6 ± 0.7	94.0 ± 0.5	82.2 ± 0.3	89.7 ± 0.3	91.5 ± 0.4
$TopLSTMs_{ens}$	82.1 ± 0.4	94.4 ± 0.5	82.7 ± 0.3	91.1 ± 0.2	92.0 ± 0.3

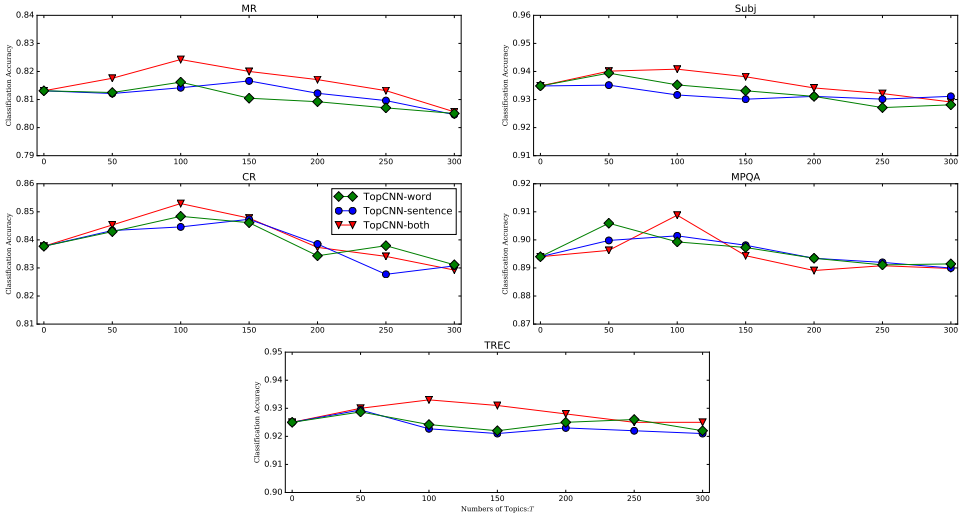

Fig. 7. Performances of our proposed three models for different number of topics on five datasets. The numbers of topics are ranged from 0 to 300 with a step size of 50. The topic number of zero corresponds to the original CNN model, and therefore these three models are the same under the zero topic number.

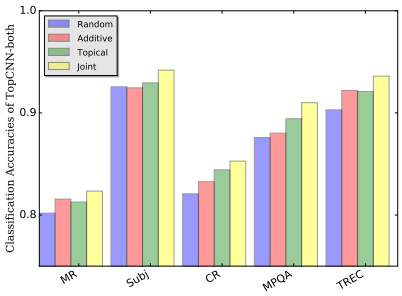

Fig. 8. Effects of different topics embeddings learning schemes on the whole five datasets. Here, we focus on $\text{TopCNN}_{\text{word\&sen}}$ model with different kinds of input topics embeddings, while other parameters are kept the same. Different colors correspond to different topic embedding learning schemes. Random, additive, topical and joint represent topics embeddings generated from random model, additive model, topical neural model and joint neural model respectively.

additive model on all datasets, despite that non-linear model has been proven to be successful in distributed representation learning. This may be due to the discrepancy of semantic spaces between topics and words. Word embeddings and topics embeddings are pre-trained independently. However, the additive model aims to derive the topic embeddings based on pre-trained word embeddings, and thus the semantic space discrepancy between words and topics may be eliminated. In addition, the adopted fine-tunning process may also be the reasons of comparable performance between additive embeddings and neural embeddings. Among the four kinds of topic embeddings, the contributions of joint neural model are the most significant with average performance gains of 1.78% and 1.25% compared to additive and topical models over these five datasets. This is because in the joint neural model, the topic and the associated word are observed together so that the embeddings for topics are learned with the word embeddings. As a result, the joint neural model yields the best performance.

V. Conclusions and Future Work

In this paper, we have introduced a topic-aware deep compositional framework and developed two models named TopCNN and TopLSTMs for sentence representation. By incorporating probability distribution derived by LDA, TopCNN and TopLSTMs are able to explore the topic-specific sense of words and sentences and further utilize such topic-specific sense jointly with general sense of word embeddings in the semantic compositional process. In addition, several effective architectures for topic embedding learning have been proposed and incorporated into our proposed framework. The experimental results have shown that $\text{TopCNN}_{\text{ens}}$ achieves competitive performances compared to the state-of-art models over five sentence classification tasks. The improved performance of TopCNN and TopLSTMs over their original models have also proven the effectiveness of our framework.

References

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[2] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 115–124.

[3] C. Brockett and W. B. Dolan, "Support vector machines for paraphrase identification and corpus construction," in *Proceedings of the 3rd International Workshop on Paraphrasing*, 2005, pp. 1–8.

[4] D. Tang, B. Qin, F. Wei, L. Dong, T. Liu, and M. Zhou, "A joint segmentation and classification framework for sentence level sentiment classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 1750–1761, Nov 2015.

[5] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 694–707, April 2016.

[6] R. E. Banchs, L. F. DHaro, and H. Li, "Adequacy-fluency metrics: Evaluating mt in the continuous space model framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 472–482, March 2015.

[7] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[8] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Advances in neural information processing systems*, 2009, pp. 1081–1088.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[10] J. Mitchell and M. Lapata, "Vector-based models of semantic composition." in *ACL*, 2008, pp. 236–244.

[11] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 151–161.

[12] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631. Citeseer, 2013, p. 1642.

[13] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models." in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, 2013, pp. 1700–1709.

[14] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1700–1709.

[15] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2014, pp. 1746–1751.

[16] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[17] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Advances in Neural Information Processing Systems*, 2014, pp. 2042–2050.

[18] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1201–1211.

[19] J. Reisinger and R. J. Mooney, "Multi-prototype vector-space models of word meaning," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 109–117.

[20] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 873–882.

[21] X. Chen, Z. Liu, and M. Sun, "A unified model for word sense representation and disambiguation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1025–1035.

[22] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space," *arXiv preprint arXiv:1504.06654*, 2015.

[23] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical word embeddings," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[24] D. Kartsaklis, M. Sadrzadeh *et al.*, "Prior disambiguation of word tensors for constructing sentence vectors." in *EMNLP*, 2013, pp. 1590–1601.

[25] J. Li and D. Jurafsky, "Do multi-sense embeddings improve natural language understanding?" *arXiv preprint arXiv:1506.01070*, 2015.

[26] J. Cheng and D. Kartsaklis, "Syntax-aware multi-sense word embeddings for deep compositional models of meaning," *arXiv preprint arXiv:1508.02354*, 2015.

[27] S. Reddy, I. P. Klapaftis, D. McCarthy, and S. Manandhar, "Dynamic and static prototype vectors for semantic composition." in *IJCNLP*, 2011, pp. 705–713.

[28] J. Cheng, D. Kartsaklis, and E. Grefenstette, "Investigating the role of prior disambiguation in deep-learning compositional models of meaning," *arXiv preprint arXiv:1411.4116*, 2014.

[29] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, 1995.

[30] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 694–707, 2016.

[31] M. Ma, L. Huang, B. Zhou, and B. Xiang, "Tree-based convolution for sentence modeling," *arXiv preprint arXiv:1507.01839*, 2015.

[32] H. Zhao, Z. Lu, and P. Poupart, "Self-adaptive hierarchical sentence model," *arXiv preprint arXiv:1504.05070*, 2015.

[33] M. Zhang, Y. Zhang, and D.-T. Vo, "Gated neural networks for targeted sentiment analysis," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA. Association for the Advancement of Artificial Intelligence*, 2016.

[34] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[35] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine learning*, vol. 42, no. 1-2, pp. 177–196, 2001.

[36] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.

[37] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 2, pp. 203–208, 1996.

[38] Z. S. Harris, "Mathematical structures of language," 1968.

[39] K. Erk and S. Padó, "A structured vector space model for word meaning in context," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 897–906.

[40] K. Erk, "Vector space models of word meaning and phrase meaning: A survey," *Language and Linguistics Compass*, vol. 6, no. 10, pp. 635–653, 2012.

[41] E. Grefenstette and M. Sadrzadeh, "Experimental support for a categorical compositional distributional model of meaning," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1394–1404.

[42] M. Baroni and R. Zamparelli, "Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 1183–1193.

[43] E. Guevara, "A regression model of adjective-noun compositionality in distributional semantics," in *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*. Association for Computational Linguistics, 2010, pp. 33–37.

[44] X. Zhu, P. Sobhani, and H. Guo, "Dag-structured long short-term memory for semantic compositionality," in *Proceedings of NAACL-HLT*, 2016, pp. 917–926.

[45] R. Zhang, H. Lee, and D. Radev, "Dependency sensitive convolutional neural networks for modeling sentences and documents," in *Proceedings of NAACL-HLT*, 2016, pp. 1512–1521.

[46] Z. Wu and C. L. Giles, "Sense-aaware semantic analysis: A multi-prototype word representation model using wikipedia," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[47] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, vol. 12, pp. 1532–1543, 2014.

[48] Y. Zhang, S. Roller, and B. Wallace, "Mgnc-cnn: A simple approach to exploiting multiple word embeddings for sentence classification," *arXiv preprint arXiv:1603.00968*, 2016.

[49] W. Yin and H. Schütze, "Multichannel variable-size convolution for sentence classification," *arXiv preprint arXiv:1603.04513*, 2016.

[50] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22nd international conference on World Wide Web*.

[51] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[52] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[53] M. D. Zeiler, "Adadelta: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[54] B. Pang and L. Lee, "A sentimen- tal education: Sentiment analysis using subjectivity sum- marization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, pp. 115–124.

[55] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[56] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language resources and evaluation*, vol. 39, no. 2-3, pp. 165–210, 2005.

[57] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.

[58] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 2012, pp. 90–94.

[59] S. Wang and C. Manning, "Fast dropout training," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 118–126.

[60] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents." in *ICML*, vol. 14, 2014, pp. 1188–1196.

[61] K.-i. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural networks*, vol. 6, no. 6, pp. 801–806, 1993.

[62] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[63] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

Rui Zhao received the BEng in Measurement and Control from Southeast University, Nanjing, China, in 2012. He is currently pursuing the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

His current research interests include text mining and machine learning.

Kezhi Mao received the BEng degree from the Jinan University, Jinan, China in 1989, the MEng degree from Northeastern University, Shenyang, China in 1992, and the PhD degree from the University of Sheffield, Sheffield, U.K. in 1998. He was a lecturer at Northeastern University from March 1992 to May 1995, a research associate at the University of Sheffield from April 1998 to September 1998, a research fellow at the Nanyang Technological University, Singapore, from September 1998 to May 2001, an assistant professor at the School of Electrical and Electronic Engineering, Nanyang Technological University from June 2001 to Sept 2005. He has been an associate professor since October 2005. His areas of interests include computational intelligence, pattern recognition, text mining, and knowledge extraction, cognitive science, and big data and text analytics.