Topic Memory Networks for Short Text Classification

Jichuan Zeng[1]*, **Jing Li**[2]†, **Yan Song**[2], **Cuiyun Gao**[1], **Michael R. Lyu**[1], **Irwin King**[1]
[1] Department of Computer Science and Engineering
The Chinese University of Hong Kong, HKSAR, China
[2] Tencent AI Lab, Shenzhen, China
[1] {jczeng,cygao,lyu,king}@cse.cuhk.edu.hk
[2] {ameliajli,clksong}@tencent.com

Abstract

Many classification models work poorly on short texts due to data sparsity. To address this issue, we propose *topic memory networks* for short text classification with a novel topic memory mechanism to encode latent topic representations indicative of class labels. Different from most prior work that focuses on extending features with external knowledge or pre-trained topics, our model jointly explores topic inference and text classification with memory networks in an end-to-end manner. Experimental results on four benchmark datasets show that our model outperforms state-of-the-art models on short text classification, meanwhile generates coherent topics.

1 Introduction

Short texts have become an important form for individuals to voice opinions and share information on online platforms. A large body of daily-generated contents, such as tweets, web search snippets, news feeds, and forum messages, have far outpaced the reading and understanding capacity of individuals. As a consequence, there is a pressing need for automatic language understanding techniques for processing and analyzing such texts (Zhang et al., 2018). Among those techniques, text classification is a critical and fundamental one proven to be useful in various downstream applications, such as text summarization (Hu et al., 2015), recommendation (Zhang et al., 2012), and sentiment analysis (Chen et al., 2017).

Although many classification models like support vector machines (SVMs) (Wang and Manning, 2012) and neural networks (Kim, 2014; Xiao and Cho, 2016; Joulin et al., 2017) have demonstrated their success in processing formal and well-edited texts, such as news articles (Zhang et al., 2015b), their performance is inevitably compromised when directly applied to short and informal online texts. This inferior performance is attributed to the severe data sparsity nature of short texts, which results in the limited features available for classifiers (Phan et al., 2008). To alleviate the data sparsity problem, some approaches exploit knowledge from external resources like Wikipedia (Jin et al., 2011) and knowledge bases (Lucia and Ferrari, 2014; Wang et al., 2017a). These approaches, however, rely on a large volume of high-quality external data, which may be unavailable to some specific domains or languages (Li et al., 2016a).

Training instances
R_1: [SuperBowl] I'll do anything to see the **Steelers** win.
R_2: [New.Music.Live] Please give **wristbands**, she have major **Bieber** Fever.
Test instance
S: [New.Music.Live] I will do anything for **wristbands**, gonna tweet till I win.

Table 1: Tweet examples for classification. R_i denotes the i-th training instance; S denotes a test instance. [class] is the ground-truth label. **Bold** words are indicative of an instance's class label.

To illustrate the difficulties in classifying short texts, we take the tweet classification in Table 1 as an example. In the test instance S, only given the 11 words it contains, it is difficult to understand why its label is New.Music.Live. Without richer context, classifiers are likely to classify S into the same category as the training instance R_1, which happens to share many words with S, in spite of the different categories they belong to,[1] rather than R_2, which only shares the word "*wristbands*" with S. Under this circumstance, how might we enrich the context of these short texts? If looking at R_2, we can observe that the semantic meaning of "*wristbands*" can be extended from its co-

* This work was mainly conducted when Jichuan Zeng was an intern in Tencent AI Lab.

† Jing Li is the corresponding author.

[1] R_1 is about SuperBowl, the annual championship game of the National Football League. R_2 and S are both about New.Music.Live, the flagship live music show.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3120–3131
Brussels, Belgium, October 31 - November 4, 2018.

occurrence with "*Bieber*", which is highly indicative of New.Music.Live.[2] Such relation can further help in recognizing the word "*wristbands*" to be important when classifying the test instance S.

Motivated by the above-mentioned observations, we present a novel neural framework, named as *topic memory networks* (TMN), for short text classification that does not rely on external knowledge. Our model can identify the indicative words for classification, e.g., "*wristbands*" in S, via jointly exploiting the document-level word co-occurrence patterns, e.g., "*wristbands*" and "*Bieber*" in R_2. To be more specific, built upon the success of neural topic models (Srivastava and Sutton, 2017; Miao et al., 2017), our model is capable of discovering *latent topics*[3], which can capture the co-occurrence of words in document level. To employ the *latent topics* for short text classification, we propose a novel *topic memory mechanism*, which is inspired by memory networks (Weston et al., 2014; Graves et al., 2014), that allows the model to put attention upon the indicative latent topics useful to classification. With such corpus-level latent topic representations, each short text instance is enriched, which thus helps alleviate the data sparsity issues.

In prior research, though the effects of topic models for short text classification have been explored (Phan et al., 2008; Ren et al., 2016), existing methods tend to use pre-trained topics as features. To the best of our knowledge, our model is the first to encode latent topic representations via memory networks for short text classification, which allows joint inference of latent topics.

To evaluate our model, we experiment and compare it with existing methods on four benchmark datasets. Experimental results indicate that our model outperforms state-of-the-art counterparts on short text classification. The quantitative and qualitative analysis illustrate the capability of our model in generating topic representations that are meaningful and indicative of different categories.

2 Topic Memory Networks

In this section, we describe our topic memory networks (TMN), whose overall architecture is shown in Figure 1. There are three major components: (1) a neural topic model (NTM) to induce latent topics (described in Section 2.1), (2) a topic memory mechanism that maps the inferred latent topics to classification features (described in Section 2.2), and (3) a text classifier, which produces the final classification labels for instances. These three components can be updated simultaneously via a joint learning process, which is introduced in Section 2.3. In particular, for the classifier, our TMN framework allows the combination of multiple options, e.g., CNN and RNN, which can be determined by the specific application scenario.

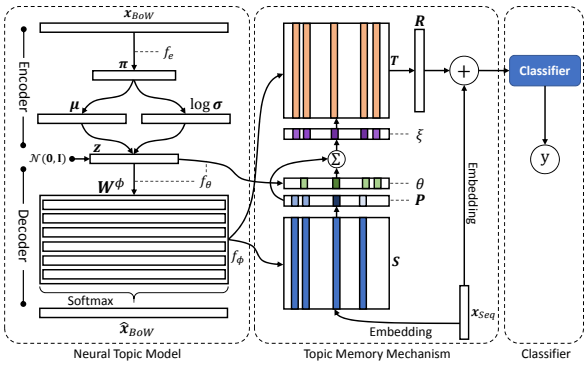


Figure 1: The overall framework of our topic memory networks. The dotted boxes from left to right show the neural topic model, the topic memory mechanism, and the classifier. Here the classifier allows multiple options and the details are left out.

Formally, given $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M\}$ as the input with M short text instances, each instance $\mathbf{x}$ is processed into two representations: bag-of-words (BoW) term vector $\mathbf{x}_{BoW} \in \mathbb{R}^V$ and word index sequence vector $\mathbf{x}_{Seq} \in \mathbb{R}^L$, where V is the vocabulary size and L is the sequence length. $\mathbf{x}_{BoW}$ is fed into the neural topic model to induce latent topics. Such topics are further matched with the embedded $\mathbf{x}_{Seq}$ to learn classification features in the topic memory mechanism. Then, the classifier concatenates the representations produced by the topic memory mechanism and the embedded $\mathbf{x}_{Seq}$ to predict the classification label y for $\mathbf{x}$.

2.1 Neural Topic Model

Our topic model is inspired by neural topic model (NTM) (Miao et al., 2017; Srivastava and Sutton, 2017) that induces latent topics in neural networks. NTM is based on variational auto-encoder (VAE) (Kingma and Welling, 2013), involved with a continuous latent variable $\mathbf{z}$ as an intermediate representation. Here in NTM, the latent variable

[2] Justine Bieber was on New.Music.Live in 2011. There was a business activity for this event that gave free wristbands to fans if they supported Bieber on Twitter.

[3] Latent topics are the distributional clusters of words that frequently co-occur in some of the instances instead of widely appearing throughout the corpus (Blei et al., 2003).

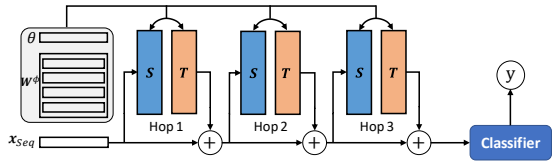


Figure 2: Topic memory network with three hops.

$\mathbf{z} \in \mathbb{R}^K$, where K denotes the number of topics. In the following, we describe the generation and the inference of the model in turn.

NTM Generation. Similar to LDA-style topic models, we assume $\mathbf{x}$ having a topic mixture θ represented as a K-dimensional distribution, which is generated via Gaussian softmax construction (Miao et al., 2017). Each topic k is represented by a word distribution ϕ_k over the vocabulary. Specifically, the generation story for $\mathbf{x}$ is:

- Draw latent variable $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$
- $\theta = \text{softmax}(f_\theta(\mathbf{z}))$
- For the n-th word in $\mathbf{x}$:
  - Draw word $w_n \sim \text{softmax}(f_\phi(\theta))$

where $f_*(\cdot)$ is a neural perceptron that linearly transforms inputs, activated by a non-linear transformation. Here we use rectified linear units (ReLUs) (Nair and Hinton, 2010) as activate functions. The prior parameters of $\mathbf{z}$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, are estimated from the input data and defined as:

$$\boldsymbol{\mu} = f_\mu(f_e(\mathbf{x}_{BoW})), \log \boldsymbol{\sigma} = f_\sigma(f_e(\mathbf{x}_{BoW})) \tag{1}$$

Note that NTM is based on VAE, where an encoder estimates the prior parameters and a decoder describes the generation story. Compared with the basic VAE, NTM includes the additional distributional vectors θ and ϕ, which can yield latent topic representations and thus ensuring their better interpretability in learning process (Miao et al., 2017).

NTM Inference. In NTM, we use variational inference (Blei et al., 2016) to approximate a posterior distribution over $\mathbf{z}$ given all the instances. The loss function of NTM is defined as

$$\mathcal{L}_{NTM} = D_{KL}(q(\mathbf{z}) \,||\, p(\mathbf{z} \,|\, \mathbf{x})) - \mathbb{E}_{q(\mathbf{z})}[p(\mathbf{x} \,|\, \mathbf{z})] \tag{2}$$

the negative of variational lower bound, where $q(\mathbf{z})$ is a standard Normal prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$. $p(\mathbf{z} \,|\, \mathbf{x})$ and $p(\mathbf{x} \,|\, \mathbf{z})$ are probabilities to describe encoding and decoding processes, respectively.[4] Due to the space limitation, we leave out the derivation details and refer the readers to Miao et al. (2017).

[4]In implementation, to smooth the gradients, we apply reparameterization on $\mathbf{z}$ following previous work (Kingma and Welling, 2013; Rezende et al., 2014).

2.2 Topic Memory Mechanism

We exploit a topic memory mechanism to map the latent topics produced by NTM (described in Section 2.1) to the features for classification. Inspired by memory networks (Weston et al., 2014; Sukhbaatar et al., 2015), we design two memory matrices, a source memory $\boldsymbol{S}$ and a target memory $\boldsymbol{T}$, both of which are in $K \times E$ size (K for the number of topics and E for the pre-defined size of word embeddings). $\boldsymbol{S}$ and $\boldsymbol{T}$ are produced by two ReLU-actived neural perceptrons, both taking the topic-word weight matrix $\boldsymbol{W}^\phi \in \mathbb{R}^{K \times V}$ as inputs. Recall that in NTM, we use $f_\phi(\cdot)$ to compute the word distributions given θ. $\boldsymbol{W}^\phi$ is the kernel weight matrix of $f_\phi(\cdot)$, where $\boldsymbol{W}^\phi_{k,v}$ represents the importance of the v-th word in reflecting the k-th topic. Assuming $\boldsymbol{U}$ as the embedded $\mathbf{x}_{Seq}$ (word sequence form of $\mathbf{x}$), in source memory, we compute the match between the k-th topic and the embedding of the l-th word in $\mathbf{x}_{Seq}$ by

$$\boldsymbol{P}_{k,l} = \text{sigmoid}(\mathbf{W}^s[\boldsymbol{S}_k; \boldsymbol{U}_l] + b^s) \tag{3}$$

where $[\mathbf{x}; \mathbf{y}]$ denotes the merge of $\mathbf{x}$ and $\mathbf{y}$, and we use concatenation operation here (Dou, 2017; Chen et al., 2017). $\mathbf{W}^s$ and b^s are parameters to be learned. To further combine the instance-topic mixture θ with $\boldsymbol{P}$, we define the integrated memory weights as

$$\xi_k = \theta_k + \gamma \sum_l \boldsymbol{P}_{k,l} \tag{4}$$

where γ is the pre-defined coefficient. Then, in target memory, via weighting target memory matrix $\boldsymbol{T}$ with ξ, we obtain the output representation $\boldsymbol{R}$ of the topic memory mechanism:

$$\boldsymbol{R}_k = \xi_k \boldsymbol{T}_k \tag{5}$$

The concatenation of $\boldsymbol{R}$ and $\boldsymbol{U}$ (embedded $\mathbf{x}_{Seq}$) further serves as the features for classification.

In particular, similar to the memory networks in prior research (Sukhbaatar et al., 2015; Chen et al., 2017), our model can be extended to handle multiple computation layers (hops). As shown in Figure 2, each hop contains a source matrix and a target matrix, and different hops are stacked following the way presented in Sukhbaatar et al. (2015).

Dataset	# of labels	# of docs	Avg len per doc	Vocab size
Snippets	8	12,332	17	7,334
TagMyNews	7	32,567	8	9,433
Twitter	50	15,056	5	6,962
Weibo	50	21,944	6	10,121

Table 2: Statistics of the experimental datasets. Labels refers to class labels. Avg len per doc refers to the average count of words in each document instance.

2.3 Joint Learning

The entire TMN model integrates the three modules in Figure 1, i.e., the neural topic model, the topic memory mechanism, and the classifier, which can be updated simultaneously in one framework. In doing so, we jointly tackle topic modeling and classification, and define the loss function of the overall framework to combine the two effects as following:

$$\mathcal{L} = \mathcal{L}_{NTM} + \lambda \mathcal{L}_{CLS} \quad (6)$$

where $\mathcal{L}_{NTM}$ represents the loss of NTM and $\mathcal{L}_{CLS}$ is the cross entropy reflecting classification loss. λ is the trade-off parameter controlling the balance between topic model and classification.

3 Experiment Setup

3.1 Datasets

We conduct experiments on four short text datasets, namely, Snippets, TagMyNews, Twitter, and Weibo. Their details are described as follows.

Snippets. This dataset contains Google search snippets released by Phan et al. (2008). There are eight ground-truth labels, e.g., *health* and *sport*.

TagMyNews. We use the news titles as instances from the benchmark classification dataset released by Vitale et al. (2012).[5] This dataset contains English news from really simple syndication (RSS) feeds. Each news feed (with its title) is annotated with one from seven labels, e.g., *sci-tech*.

Twitter. This dataset is used to evaluate tweet topic classification, which is built on the dataset released by TREC2011 microblog track.[6] Following previous settings (Yan et al., 2013; Li et al., 2016a), hashtags, i.e., user-annotated topic labels in each tweet such as "*#Trump*" and "*#SuperBowl*", serve as our ground-truth class labels. Specifically, we construct the dataset with the following steps. First, we remove the tweets without hashtags. Second, we rank hashtags by their frequencies. Third, we manually remove the hashtags that cannot mark topics, such as "*#fb*" for indicating the source of tweets from Facebook, and combine the hashtags referring to the same topic, such as "*#DonaldTrump*" and "*#Trump*". Finally, we select the top 50 frequent hashtags, and all tweets containing these hashtags.

Weibo. To evaluate our model on a different language other than English, we employ a Chinese dataset with short segments of text for topic classification. This dataset is released by Li et al. (2016b) with a collection of messages posted in June 2014 on Weibo, a popular Twitter alike platform in China.[7] Similar to Twitter, Weibo allows up to 140 Chinese characters in its messages. In this Weibo dataset, each Weibo message is labeled with a hashtag as its category, and there are 50 distinct hashtag labels in total, following the same procedure performed for the Twitter dataset.

Table 2 shows the statistic information of the four datasets. Each dataset is randomly split into 80% for training and 20% for test. 20% of randomly selected training instances are used to form development set. We preprocess our English datasets, i.e., Snippets, TagMyNews, and Twitter, with *gensim tokenizer*[8] for tokenization. As to the Chinese Weibo dataset, we use FudanNLP toolkit (Qiu et al., 2013)[9] for word segmentation. In addition, for each dataset, we maintain a vocabulary built based on the training set with removal of stop words[10] and words occurring less than 3 times. The inputs of topic models $\mathbf{x}_{BoW}$ are constructed based on this vocabulary following common topic model settings (Blei et al., 2003; Miao et al., 2016). Differently, we use the raw word sequence (without words removal) for the inputs of classification $\mathbf{x}_{Seq}$ as is done in previous work of text classification (Kim, 2014; Liu et al., 2017).

3.2 Model Settings

We use pre-trained embeddings to initialize all word embeddings. For Snippets and TagMyNews

[5] http://acube.di.unipi.it/tmn-dataset/

[6] http://trec.nist.gov/data/tweets

[7] The original dataset contains conversations to enrich the context of Weibo posts, which are not considered here.

[8] https://radimrehurek.com/gensim/utils.html

[9] https://github.com/FudanNLP/fnlp

[10] https://radimrehurek.com/gensim/parsing/preprocessing.html

Models	Snippets		TagMyNews		Twitter		Weibo	
	Acc	Avg F1	Acc	Avg F1	Acc	Avg F1	Acc	Avg F1
Comparison models								
Majority Vote	0.202	0.068	0.247	0.098	0.073	0.010	0.102	0.019
SVM+BOW (Wang and Manning, 2012)	0.210	0.080	0.259	0.058	0.070	0.009	0.116	0.039
SVM+LDA (Blei et al., 2003)	0.689	0.694	0.616	0.593	0.159	0.111	0.192	0.147
SVM+BTM (Yan et al., 2013)	0.772	0.772	0.686	0.677	0.232	0.164	0.331	0.277
SVM+NTM (Miao et al., 2017)	0.779	0.776	0.664	0.654	0.261	0.177	0.379	0.348
AttBiLSTM (Zhang and Wang, 2015)	0.943	0.943	0.838	0.828	0.375	0.348	0.547	0.547
CNN (Kim, 2014)	0.944	0.944	0.843	0.843	0.381	0.362	0.553	0.550
CNN+TEWE (Ren et al., 2016)	0.944	0.944	0.846	0.846	0.385	0.368	0.537	0.532
CNN+NTM	0.945	0.945	0.844	0.844	0.382	0.365	0.556	0.556
Our models								
TMN (*Separate TM Inference*)	0.961	0.961	0.848	0.847	0.394	**0.386**	0.568	0.569
TMN (*Joint TM Inference*)	**0.964**	**0.964**	**0.851**	**0.851**	**0.397**	0.375	**0.591**	**0.589**

Table 3: Comparisons of accuracy (Acc) and average F1 (Avg F1) on four benchmark datasets. Our TMN, either with separate or joint TM inference, performs significantly better than all the comparisons ($p < 0.05$, paired t-test).

datasets, we use pre-trained `GloVe` embeddings (Pennington et al., 2014)[11]. For Twitter and Weibo datasets, we pre-train embeddings on large-scale external data with 99M tweets and 467M Weibo messages, respectively. For the number of topics, we follow previous settings (Yan et al., 2013; Das et al., 2015; Dieng et al., 2016) to set $K = 50$. For all the other hyperparameters, we tune them on the development set by grid search. For our classifier, we employ CNN in experiment because of its better performance in short text classification than its counterparts such as RNN (Wang et al., 2017a). The hidden size of CNN is set as 500. The dimension of word embedding $E = 200$. $\gamma = 0.8$ for trading off θ and $\boldsymbol{P}$, and $\lambda = 1.0$ for controlling the effects of topic model and classification. In the learning process, we run our model for 800 epochs with early-stop strategy applied (Caruana et al., 2000).

3.3 Comparison Models

For comparison, we consider a weak baseline of majority vote, which assigns the major class labels in training set to all test instances. We further compare with the widely-used baseline SVM+BOW, SVM with unigram features (Wang and Manning, 2012). We also consider other SVM-based baselines: SVM+LDA, SVM+BTM, SVM+NTM, whose features are topic distributions for instances learned by LDA (Blei et al., 2003), BTM (Yan et al., 2013), and NTM (Miao et al., 2017), respectively. In particular, BTM is one of the state-of-the-art topic models for short texts. To compare with neural classifiers, we test bidirectional long short-term memory with attention (AttBiLSTM) (Zhang et al., 2015a) and convolutional neural network (CNN) classifiers (Kim, 2014). No topic representation is encoded in these two classifiers. We also compare with the state-of-the-art short-text classifier CNN+TEWE (Ren et al., 2016), i.e., CNN classifier with topic-enriched word embeddings (TEWE), where the word embeddings are enriched by pre-trained NTM-inferred topic models. Moreover, to investigate the effectiveness of our proposed topic memory mechanism, we compare with CNN+NTM, which concatenates the representations learned by CNN and topics induced by NTM as classification features. In addition, we compare with our variant, TMN (*Separate TM Inference*), where topics are induced separately before classification, and only used for initializing the topic memory. To be consistent, our model with a joint learning process for topic modeling and classification, described in Section 2.3, is named as TMN (*Joint TM Inference*). Note that the comparison CNN-based models share the same settings as our model, and the hidden size for each direction of BiLSTM is set to 100.

4 Experimental Results

4.1 Classification Comparison

Table 3 shows the comparison on classification results, where the accuracy and average F1 scores on different classes labels are reported. We have the following observations.

• ***Topic representations are indicative features.*** On all four datasets, simply by combining topic representations into features, SVM models produce better results than the models without ex-

[11] `http://nlp.stanford.edu/data/glove.6B.zip` (200d)

Model	Snippets	TagMyNews	Twitter
LDA	0.436	0.449	0.436
BTM	0.435	0.463	0.435
NTM	0.463	0.468	0.463
TMN	**0.487**	**0.499**	**0.468**

Table 4: C_V coherence scores for topics generated by various models. Higher is better. The best result in each column is in **bold**.

ploiting topic features (*i.e.*, SVM+BOW). This observation indicates that latent topic representations captured at corpus level are helpful to alleviate the data sparsity problem in short text classification.

• ***Neural network models are effective.*** It is seen that neural models based on either CNN or AttBiLSTM yield better results than SVM. This observation shows the effectiveness of representation learning in neural networks for short texts.

• ***CNN serves as a better classifier for short texts than AttBiLSTM.*** In comparison of CNN and AttBiLSTM without taking topic features, we observe that CNN yields generally better results on all the four datasets. This is consistent with the discovery in Wang et al. (2017a), where CNN can better encode short texts than sequential models.

• ***Topic memory is useful to classification.*** By exploring topic representations in memory mechanisms, our TMN model, inferring topic models either separately or jointly with classification, significantly outperform the best comparison models on each of the four datasets. Particularly, when compared with CNN+TEWE and CNN+NTM, both concatenating topics as part of the features, the results yielded by TMN are better. This demonstrates the effectiveness of topic memory to learn indicative topic representations for short text classification.

• ***Jointly inferring latent topics is effective to text classification.*** In comparison between two TMN variants, TMN (*Joint TM Inference*) produces better classification results, though large margin improvements are not observed on the three English datasets, i.e., TagMyNews, Snippets, and Twitter. This may be because the classifiers do not rely too much on high-quality latent topics, since other features may be sufficient to indicate the labels, e.g., word positions in the instance. As a result, better topic models, learned via jointly induced with classification, may not provide richer information for classification. Nevertheless, we notice that on

LDA	mubarak bring run obama democracy speech believe regime power bowl
BTM	mubarak egypt push internet people government phone hosni need son
NTM	mubarak people egyptian egypt stay tomorrow protest news phone protester
TMN	mubarak protest protester tahrir square egyptian al jazeera repo cairo

Table 5: Top 10 representative terms of the sample latent topics discovered by various topic models from Twitter dataset. We interpret the topics as "*Egyptian revolution of 2011*" according to their word distributions. Non-topic words are wave-underlined and in blue, and off-topic words are underlined and in red.

Chinese Weibo dataset, the jointly trained topic model improves the accuracy and average F1 by 2.3% and 2.0%, respectively. It may result from the prevalence of word order misuse in informal Weibo messages. This mis-order phenomenon is common in Chinese and generally does not affect understanding. The rich information conveyed by Chinese characters are capable of indicating semantic meanings of words even without correct orders (Qin et al., 2016; Wang et al., 2017b). As a result, the CNN classifier, which encodes orders of words, may also bring such mis-order noise to classification. For these instances with mis-ordered words, a better topic model that learns text instances as unordered words, provides useful representations that compensate the loss of information in word orders and in turn improves the performance of text classification.

4.2 Topic Coherence Comparison

In Section 4.1, we find that TMN can significantly outperform comparison models on short text classification. In this section, we study whether jointly learning topic models and classification can be helpful in producing coherent and meaningful topics. We use the C_V metric (Röder et al., 2015) computed by Palmetto toolkit[12] to evaluate the topic coherence, which has been shown to give the closest scores to human evaluation compared to other widely-used topic coherence metrics like NPMI (Bouma, 2009). Table 4 shows the comparison results of LDA, BTM, NTM, and TMN on the three English datasets.[13] Note that we do not report C_V scores for Chinese Weibo dataset as the Palmetto toolkit cannot process Chinese topics.

[12] https://github.com/dice-group/Palmetto

[13] In the rest of this paper, without otherwise indicated, TMN is used as a short form for TMN (*Joint TM Inference*).

# of Hops	Snippets	TagMyNews	Twitter	Weibo
TMN-1H	0.958	0.841	0.382	0.568
TMN-2H	**0.964**	0.843	0.383	0.578
TMN-3H	0.962	0.845	0.384	0.581
TMN-4H	0.961	0.846	0.389	0.582
TMN-5H	0.960	**0.851**	**0.397**	**0.591**
TMN-6H	0.958	0.848	0.388	0.579

Table 6: The impact of the # of hops on accuracy.

As can be seen, TMN yields higher C_V scores by large margins than all others in comparison. This indicates that jointly exploring classification would be effective in producing coherent topics. The reason is that the supervision from classification labels can guide unsupervised topic models in discovering meaningful and interpretable topics. We also observe that NTM produces better results than LDA and BTM, which implies the effectiveness of inducing topic models by neural networks.

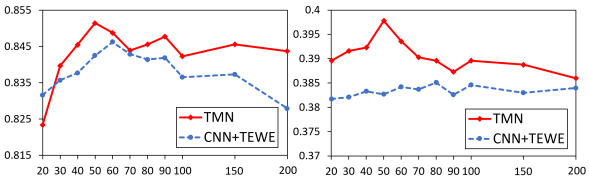
To further analyze the quality of yielded topics, Table 5 shows the top 10 words of the sample latent topics reflecting "*Egyptian revolution of 2011*" discovered by various models. We find that LDA yields off-topic word "*bowl*". For the results of BTM and NTM, though we do not find off-topic words, non-topic words like "*need*" and "*stay*" are included.[14] The topic generated by TMN appears to be the best, which presents indicative words like "*tahrir*" and "*cairo*", for the event.

4.3 Results with Varying Hyperparameters

We further study the impact of two important hyperparameters in TMN, i.e., the hop number and the topic number, which will be discussed in turn.

Impact of Hop Numbers. Recall that Figure 2 shows the capacity of TMN in combining multiple hops. Here we analyze the effects of hop numbers on the accuracy of TMN. Table 6 reports the results, where NH refers to using N hops ($N = 1, 2, ..., 6$). As can be seen, generally, TMN with 5 hops achieves the best accuracy on most datasets except for Snippets dataset. We also observe that, although within a particular range, more hops can produce better accuracy, the increasing trends are not always monotonic. For example, TMN-6H always exhibits lower accuracy than TMN-5H. This observation implies that the overall representation ability of TMN is enhanced as the increasing complexity of the model via combining more hops.

[14]Off-topic words are more likely to be interpreted to reflect other topics. Non-topic words cannot clearly indicate the corresponding topic.



(a) TagMyNews (b) Twitter

Figure 3: The impact of topic numbers, where the horizontal axis shows the number of topics and the vertical axis shows the accuracy.

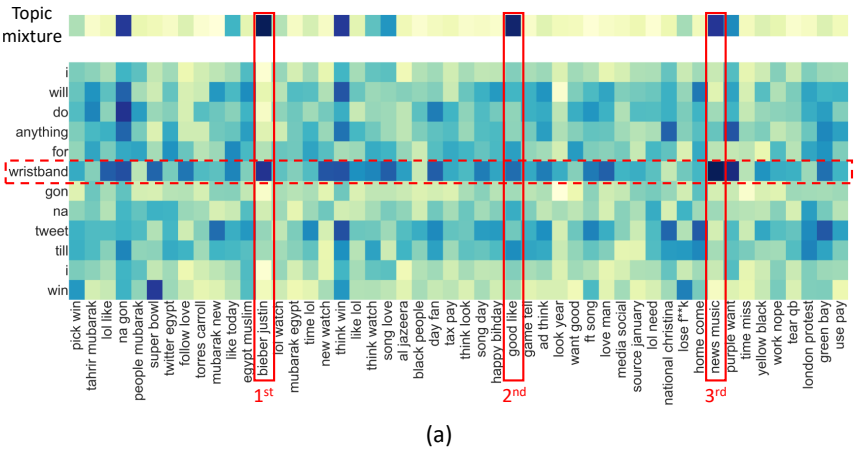
However, this enhancement will reach saturation when the hop number exceeds a threshold, which is 5 hops for most datasets in our experiment.

Impact of Topic Numbers. Figure 3 shows the accuracy of TMN and CNN+TEWE (the best comparison model in Table 3) given varying K, the number of topics on TagMyNews and Twitter datasets.[15] As we can see, the curves of all the models are not monotonic and the best accuracy is achieved given a particular number of topics, e.g., K=50 for TMN on TagMyNews dataset. When comparing different curves, we observe that TMN yields consistently better accuracy than CNN+TEWE, a comparison model shown in Table 3, which demonstrates the robust performance of TMN over varying number of topics.

4.4 A Case Study on Topic Memory

Section 4.1 demonstrates the effectiveness of using topic memory on short text classification. To further understand why, in this section, we use the test instance S in Table 1 to analyze what the information captured by topic memory is indicative of class labels. Recall that the label of S, which should be New.Music.Live, can be indicated by containing word "*wristbands*" and the collocation of "*wristbands*" and "*Bieber*" in training instance R_2 labeled New.Music.Live. Figure 4 shows the heatmaps of the weight matrix $\boldsymbol{P}$ in topic memory and the topic mixture θ captured by NTM for instance S. As can be seen, the top 3 words for the latent topic with the largest value in θ are "*bieber*", "*justine*", and "*tuesday*", which can effectively indicate the class label of S to be *New.Music.Live* because Justine Bieber was there on Tuesday. Interestingly, S contains none of the top three words. The latent semantic relations of S and these words are purely uncovered by the co-occurrence of words in S with other instances in

[15]We observe similar distributions on Snippets and Weibo.



(a)

1st Topic	2nd Topic	3rd Topic
bieber	good	news
justin	like	music
tuesday	fan	forever
newmusiclive	look	change
think	tweet	right
thought	thing	attack
jb	know	time
new	rumble	wing
music	na	aistdirect
live	right	films

(b)

Figure 4: Topic memory visualization for test instance S shown in Table 1. (a) Heatmaps of topic mixture θ (the upper one) and topic memory weight matrix $\boldsymbol{P}$ (the lower one) illustrating the relevance between the words of S (left) and the learned topics (bottom, with top-2 words displayed). The red dotted rectangle indicates the representation for "*wristband*", the topical word in S. The red rectangles with solid frames indicates the 3 most relevant topics ordered by θ. (b) Top-10 words of these topics indicated by ϕ.

the corpus, which further shows the benefit of using latent topics for alleviating the sparsity in short texts. We also observe that topic memory learns different representations for topical word "*wristband*", highly indicating instance label, and background words, such as "*i*" and "*for*". This explains why topic memory is effective to classification.

4.5 Error Analysis

In this section, we take our classification results on TagMyNews dataset as an example to analyze our errors. We observe that one major type of incorrect prediction should be ascribed to the polysemy phenomenon. For example, the instance "*NBC gives 'the voice' post super bowl slot*" should be categorized as *entertainment*. However, failing to understand the particular meaning of "*the voice*" here as the name of a television singing competition, our model mistakenly categorizes this instance as sport because of the occurrence "*super bowl*". In future work, we would exploit context-sensitive topical word embeddings (Witt et al., 2016), which is able to distinguish the meanings of the same word in different contexts. Another main error type comes from the failure to capture phrase-level semantics. Taking "*On the merits of face time and living small*" as an example, without understanding "*face time*" as a phrase, our model wrongly predicts its category as business instead of its correct label as sci_tech. Such errors can be reduced by enhancing our NTM to phrase discovery topic models (Lindsey et al., 2012; He, 2016), which is worthy exploring in future work.

5 Related Work

Our work mainly builds on two streams of prior work: short text classification and topic models.

Short Text Classification. In the line of short text classification, most work focuses on alleviating the severe sparsity issues in short texts (Yan et al., 2013). Some previous efforts encode knowledge from external resource (Jin et al., 2011; Lucia and Ferrari, 2014; Wang et al., 2017a; Ma et al., 2018). Instead, our work learns effective representations only from internal data. For some specific classification tasks, such as sentiment analysis, manually-crafted features are designed to fit the target task (Pak and Paroubek, 2010; Jiang et al., 2011). Distinguished from them, we employ deep learning framework for representation learning, which requires no feature engineering process and thus ensures its general applicability to diverse classification scenarios. In comparison with the established classifiers applying deep learning methods (dos Santos and Gatti, 2014; Lee and Dernoncourt, 2016), our work differs from them in the leverage of corpus-level latent topic representations for alleviating data sparsity issues. In existing classification models using topic features, pre-trained topic mixtures are leveraged as part of features (Phan et al., 2008; Ren et al., 2016; Chen et al., 2017). Differently, our model encodes topic representations in a memory mechanism where topics are induced jointly with text classification in an end-to-end manner.

Topic Models. Well-known topic models, e.g., probabilistic latent semantic analysis (pLSA) (Hofmann, 1999) and latent Dirichlet allocation (LDA) (Blei et al., 2003), have shown advantages in capturing effective semantic representations, and proven beneficial to varying downstream applications, such as summarization (Haghighi and Vanderwende, 2009) and recommendation (Zeng et al., 2018; Bai et al., 2018). For short text data, topic model variants have been proposed to reduce the effects of sparsity issues on topic modeling, such as biterm topic model (BTM) (Yan et al., 2013) and LeadLDA (Li et al., 2016b). Recently, owing to the popularity of variational auto-encoder (VAE) (Kingma and Welling, 2013), it is able to induce latent topics in neural networks, namely, neural topic models (NTM) (Miao et al., 2017; Srivastava and Sutton, 2017). Although the concept of NTM has been mentioned earlier in Cao et al. (2015), their model is based on matrix factorization. Differently, VAE-style NTM (Srivastava and Sutton, 2017; Miao et al., 2017) follows the LDA fashion as probabilistic generative models, which is easy to interpret and extend. The NTM in our framework is in VAE-style, whose effects on short text classification serve as the key focus of our work.

6 Conclusion

We have presented topic memory networks that exploit corpus-level topic representations with a topic memory mechanism for short text classification. The model alleviates data sparsity issues via jointly learning latent topics and text categories. Empirical comparisons with state-of-the-art models on four benchmark datasets have demonstrated the validity and effectiveness of our model, where better results have been achieved on both short text classification and topic coherence evaluation.

Acknowledgements

This work is partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14208815 and No. CUHK 14234416 of the General Research Fund), and Microsoft Research Asia (2018 Microsoft Research Asia Collaborative Research Award). We thank Shuming Shi, Dong Yu, Tong Zhang, Baolin Peng, Haoli Bai, and the three anonymous reviewers for the insightful suggestions on various aspects of this work.

References

Haoli Bai, Zhuangbin Chen, Michael R. Lyu, Irwin King, and Zenglin Xu. 2018. Neural Relational Topic Model for Scientific Article Analysis. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*. ACM.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2016. Variational Inference: A Review for Statisticians. *CoRR*, abs/1601.00670.

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

Gerlof Bouma. 2009. Normalized (Pointwise) Mutual Information in Collocation Extraction. *Proceedings of GSCL*, pages 31–40.

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. 2015. A Novel Neural Topic Model and Its Supervised Extension. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, Texas, USA, pages 2210–2216.

Rich Caruana, Steve Lawrence, and C. Lee Giles. 2000. Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, an"d Early Stopping. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems, NIPS 2000*, Denver, CO, USA, pages 402–408.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, Copenhagen, Denmark, pages 452–461.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for Topic Models with Word Embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, Volume 1: Long Papers*, Beijing, China, pages 795–804.

Adji B. Dieng, Chong Wang, Jianfeng Gao, and John William Paisley. 2016. TopicRNN: A Recurrent Neural Network with Long-Range Semantic Dependency. *CoRR*, abs/1611.01702.

Zi-Yi Dou. 2017. Capturing User and Product Information for Document Level Sentiment Analysis with Deep Memory Network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, Copenhagen, Denmark, pages 521–526.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural Turing Machines. *CoRR*, abs/1410.5401.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2009*, pages 362–370, Boulder, Colorado, USA.

Yulan He. 2016. Extracting Topical Phrases from Clinical Documents. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA, pages 2957–2963.

Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99*, Berkeley, CA, USA, pages 50–57.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A Large Scale Chinese Short Text Summarization Dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, Lisbon, Portugal, pages 1967–1972.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter Sentiment Classification. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, Portland, Oregon, USA, pages 151–160.

Ou Jin, Nathan Nan Liu, Kai Zhao, Yong Yu, and Qiang Yang. 2011. Transferring Topical Knowledge from Auxiliary Long Texts for Short Text Clustering. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011*, Glasgow, United Kingdom, pages 775–784.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, A meeting of SIGDAT, a Special Interest Group of the ACL*, Doha, Qatar, pages 1746–1751.

Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *CoRR*, abs/1312.6114.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. In *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016*, San Diego California, USA, pages 515–520.

Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016a. Topic Modeling for Short Texts with Auxiliary Word Embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016*, Pisa, Italy, pages 165–174.

Jing Li, Ming Liao, Wei Gao, Yulan He, and Kam-Fai Wong. 2016b. Topic Extraction from Microblog Posts Using Conversation Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Volume 1: Long Papers*, Berlin, Germany.

Robert V. Lindsey, William Headden, and Michael Stipicevic. 2012. A Phrase-Discovering Topic Model Using Hierarchical Pitman-Yor Processes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012*, Jeju Island, Korea, pages 214–222.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-task Learning for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Volume 1: Long Papers*, Vancouver, Canada, pages 1–10.

William Lucia and Elena Ferrari. 2014. EgoCentric: Ego Networks for Knowledge-based Short Text Classification. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014*, Shanghai, China, pages 1079–1088.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering Discrete Latent Topics with Neural Variational Inference. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, Sydney, NSW, Australia, pages 2410–2419.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural Variational Inference for Text Processing. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016*, New York City, NY, USA, pages 1727–1736.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Haifa, Israel, pages 807–814.

Alexander Pak and Patrick Paroubek. 2010. Twitter Based System: Using Twitter for Disambiguating Sentiment Ambiguous Adjectives. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010*, Uppsala University, Uppsala, Sweden, pages 436–439.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, A meeting of SIGDAT, a Special Interest Group of the ACL*, Doha, Qatar, pages 1532–1543.

Xuan Hieu Phan, Minh Le Nguyen, and Susumu Horiguchi. 2008. Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008*, Beijing, China, pages 91–100.

Zengchang Qin, Yonghui Cong, and Tao Wan. 2016. Topic Modeling of Chinese Language beyond a Bag-of-Words. *Computer Speech & Language*, 40:60–78.

Xipeng Qiu, Qi Zhang, and Xuanjing Huang. 2013. FudanNLP: A Toolkit for Chinese Natural Language Processing. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016. Improving Twitter Sentiment Classification Using Topic-Enriched Multi-Prototype Word Embeddings. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA, pages 3038–3044.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, Beijing, China, pages 1278–1286.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the Space of Topic Coherence Measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015*, Shanghai, China, pages 399–408.

Cícero Nogueira dos Santos and Maira Gatti. 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, COLING 2014*, Dublin, Ireland, pages 69–78.

Akash Srivastava and Charles Sutton. 2017. Autoencoding Variational Inference For Topic Models. *arXiv preprint arXiv:1703.01488*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-To-End Memory Networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems, NIPS 2015*, Montreal, Quebec, Canada, pages 2440–2448.

Daniele Vitale, Paolo Ferragina, and Ugo Scaiella. 2012. Classification of Short Texts by Deploying Topical Annotations. In *Advances in Information Retrieval - 34th European Conference on IR Research, ECIR 2012*, Barcelona, Spain, pages 376–387.

Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017a. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, Melbourne, Australia, pages 2915–2921.

Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2017b. Exploiting Word Internal Structures for Generic Chinese Sentence Representation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, Copenhagen, Denmark, pages 298–303.

Sida I. Wang and Christopher D. Manning. 2012. Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, ACL 2012, Volume 2: Short Papers*, Jeju Island, Korea, pages 90–94.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory Networks. *CoRR*, abs/1410.3916.

Nils Witt, Christin Seifert, and Michael Granitzer. 2016. Explaining Topical Distances Using Word Embeddings. In *27th International Workshop on Database and Expert Systems Applications, DEXA 2016 Workshops*, Porto, Portugal, pages 212–217.

Yijun Xiao and Kyunghyun Cho. 2016. Efficient Character-level Document Classification by Combining Convolution and Recurrent Layers. *CoRR*, abs/1602.00367.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A Biterm Topic Model for Short Sexts. In *22nd International World Wide Web Conference, WWW 2013*, Rio de Janeiro, Brazil, pages 1445–1456.

Xingshan Zeng, Jing Li, Lu Wang, Nicholas Beauchamp, Sarah Shugars, and Kam-Fai Wong. 2018. Microblog conversation recommendation via joint modeling of topics and discourse. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*, New Orleans, Louisiana, USA, pages 375–385.

Dongxu Zhang and Dong Wang. 2015. Relation Classification via Recurrent Neural Network. *CoRR*, abs/1508.01006.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015a. Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, PACLIC 29*, Shanghai, China.

Weinan Zhang, Dingquan Wang, Gui-Rong Xue, and Hongyuan Zha. 2012. Advertising Keywords Recommendation for Short-Text Web Pages Using Wikipedia. *ACM TIST*, 3(2):36:1–36:25.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015b. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems, NIPS 2015*, Montreal, Quebec, Canada, pages 649–657.

Yingyi Zhang, Jing Li, Yan Song, and Chengzhi Zhang. 2018. Encoding conversation context for neural keyphrase extraction from microblog posts. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*, New Orleans, Louisiana, USA, pages 1676–1686.