Bi-level Masked Multi-scale CNN-RNN Networks for Short Text Representation

Qian Li*, Qiang Wu*, Chengzhang Zhu† and Jian Zhang*
*Global Big Data Technologies Centre, University of Technology Sydney, Australia
Email: Qian.Li-7@student.uts.edu.au, Qiang.Wu@uts.edu.au, Jian.Zhang@uts.edu.au
†Advanced Analytics Institute, University of Technology Sydney, Australia
Email: kevin.zhu.china@gmail.com

***Abstract*—Representing short text is becoming extremely important for a variety of valuable applications. However, representing short text is critical yet challenging because it involves lots of informal words and typos (i.e. the noise problem) but only few vocabularies in each text (i.e. the sparsity problem). Most of existing work on representing short text relies on noise recognition and sparsity expansion. However, the noises in short text are with various forms and changing fast, but, most of the current methods may fail to adaptively recognize the noise. Also, it is hard to explicitly expand a sparse text to a high-quality dense text. In this paper, we tackle the noise and sparsity problems in short text representation by learning multi-grain noise-tolerant patterns and then embedding the most significant patterns in a text as its representation. To achieve this goal, we propose a bi-level multi-scale masked CNN-RNN network to embed the most significant multi-grain noise-tolerant relations among words and characters in a text into a dense vector space. Comprehensive experiments on five large real-world data sets demonstrate our method significantly outperforms the state-of-the-art competitors.**

1. Introduction

The increasing of the mobile Internet and social media generates a large number of short texts, which are very short and always with informal words and typos. Effectively representing short texts into a vector space that embeds the semantic meaning of the texts is valuable and widely required by a variety of applications, such as sentiment analysis [1], churn detection [2], question retrieval [3], and personalized recommendation [4], [5].

However, short text representation is very challenging compared with formal long text representation because of its two essential characteristics: *noise* and *sparsity* [6], [7], where *noise* refers to informal words and typos, and *sparsity* means the rare number of vocabularies in a text (because of the short length limitation). As a result, most of current text representation methods may fail to represent short text. For example, the word2vec-based methods, such as [8], [9], need to look up pre-trained word representations but the informal words and typos may not appear in the training data. For another example, when facing sparsity, the term frequency - inverse document frequency (tf-idf) and the bag-of-words method will generate a very sparse representation for a short text, i.e. most of entries in the representation vector are 0, which may be meaningless for downstream learning tasks because distances between all texts are equal.

Recently, several methods have been proposed for short text representation. They achieved advanced performance by either reducing noise or alleviating sparsity. Regarding noise reduction, most of the current methods [6], [7], [10], [11], [12] first recognize noise by looking up pre-defined noise sets or adopting frequency-based detection models. Then, they reduce the effects of the recognized noise by re-weighting or ensemble strategies. However, two problems may arise in their recognition process: (1) pre-defined noise sets may not fully cover all noise; and (2) frequency-based detection models may fail when facing the sparsity. As a result, their unrecognized noises may still damage representation performance, even with re-weighting and ensemble strategies. Regarding sparsity alleviation, most of the current methods [13], [14], [15], [16] assume a sparse short text is generated from a latent dense document, and try to insert words into the short text according to the latent document, which is also known as expansion-based method. However, the quality of the expansion always cannot be guaranteed because (1) many short texts are independent that are not generated from the a common document; and (2) most of these methods are based on statistics which may heavily be affected by the noise in short text.

We address the above problems by learning *multi-grain noise-tolerant patterns* in texts and embedding the most significant patterns in a text into a dense vector space to form the text representation. The rationale is that one or more same noise-tolerant patterns should exist in multiple texts with the same semantic meaning, no matter how many noises in these texts and how sparse the texts are. Effectively embedding such patterns into a dense space will also avoid the meaningless sparse representation caused by text sparsity. For example, two sentences "Does anyone know how to repair?:(" and "dz ne1 knw h2 ripair?:(" have the same meaning. Although the second sentence has many informal words and typos, it has many explicit noise-tolerant patterns which are same as that in the first sentence, such as "knw" and "rpair" at character-level and ":(" symbol at word-level. These noise-tolerant patterns reflect the semantic meaning of the short texts.

Inspired by [7], we embed the multi-grain noise-tolerant patterns by a bi-level neural network, which captures the semantic relations between both words and characters with different granularities to tackle the sparsity problem. The intuition is that the semantic meaning of a short text may reflected by the relations among words and characters with different granularities as shown in the above example.

To further tolerant noise, we propose a *breaking-gathering* strategy. In the breaking stage, the strategy breaks a piece of text into all combinations of its vocabularies but keeps their ordinal information in the text. Then, at the gathering stage, it discovers the most significant patterns in these combinations as the pattern of the piece of text. Trough this process, the breaking-gathering strategy can adaptively filter noise with arbitrary form, and thus, discover the noise-tolerant patterns. For example, at the character-level, the word "know" will be broken as "konw", "kow", "knw", "kw" etc. in the breaking stage. Considering "knw" in another short text, the noise-tolerant pattern "knw" can be discovered in the gathering stage.

Based on the above analysis, we propose a **bi**-level **m**asked multi-sc**a**le **c**onvolutional and **r**ecurrent neural netw**o**rk (Bi-MACRO). Our method jointly captures multi-grain relations of words and characteristics to discover noise-tolerant patterns and embeds them as a dense vector representation for a short text. In summary, the main contributions of this work are as follows.

- *A bi-level neural network representation architecture.* The bi-level neural network representation architecture captures the semantic meaning of a text from both word-level and character-level, and it embeds the captured semantic meaning into a dense vector space that tackles the sparsity problem.
- *A masked CNN layer for adaptively noise filtering.* The masked CNN layer filters noise by a set of masks, and adaptively selects the most significant pattern by a cross-filter max pooling. Combining with the bi-level architecture, the masked CNN layer significantly reduces the noises at both word-level and character-level.
- *A multi-scale CNN-RNN structure.* Bi-MACRO uses a set of CNN with different filter sizes to capture short-term word and character relations with different granularities. The different filter sizes also induce masks with different mask locations, which fit the noise with arbitrary position. Further, the connected RNN layer captures the long-term relations between words and characters, and reduces the potential model complexity that may be caused by adopting a large filter size.

We conduct comprehensive experiments on five widely used real-world data sets, including TREC, Quora, Twitter, News and AG News, to show the short text characteristics and evaluate the performance of our proposed method. We show that the proposed Bi-MACRO method significantly outperforms three state-of-the-art competitors and two baseline methods in terms of short text representation.

2. Related Work

Recently, many efforts have been made on short text representation. We can generally classify these methods into two categories according to their focused characteristics: *noise* and *sparsity*.

To tackle the noise problem in short text representation, the most widely used method is filtering noises by looking up pre-defined noise sets. However, pre-defined noise sets are fixed and cannot comprehensively cover noises. Recently, some advanced learning methods have been proposed for the noise problem. For example, Dey et al. [6] propose a feature selection method to select noise-tolerant features from a set of designated lexical, syntactic, semantic, and pragmatic features. Boom et al. [10] propose a method that weighted sums word embeddings for a short text representation and adopts a median-based loss function to reduce the effect of noises in the weight learning process. However, the summation of word embeddings ignores the ordinal of the words, which may also determine the semantic meaning of the short text. Catering for the topic model, Li et al. [12] model the noises in a short text by a common distribution to filter out the noise influences. However, this method may equally treat the noises that have different meanings.

To tackle the sparsity, Zuo et al. [13] propose a pseudo-document-based topic model that models the topic distributions of a latent pseudo document rather than a short text. Instead of explicitly modeling the latent document, Liang et al. [14] and Li et al. [16] implicitly consist the latent document by assuming all words in a short text have a same topic. Further, Zheng et al. [17] expand the short text by adding new words that not appear in the short text with a virtual term frequency for each word, which is calculated by the posterior probability of the new word given all existing words in the text. The above methods are also known as the expansion-based methods. The expansion-based methods, also including [18], [19], [20], provide more representative information to short texts representation. Lochter et al. [15] further aggregate different expansion-based methods by an ensemble method and show a significant advance. Instead of the expansion, more recently, another paradigm tries to tackle the sparsity problem by leveraging complex relations between words and/or characters. For example, Lu et al. [21] and Shi et al. [22] consider the inter-word relations to capture the short text semantic meaning. Wang et al. [7] further integrate the explicit and implicit knowledge involved in a short text with the embedded complex relations to form the text representation. However, the above methods do not tolerant the noise in a short text.

3. Methodology

3.1. Overall Architecture

The architecture of Bi-MACRO is shown in Fig. 1. Given a short text, Bi-MACRO first represents it into a word embedding matrix and a character embedding matrix. Then, for each matrix, Bi-MACRO adopts a masked multi-scale

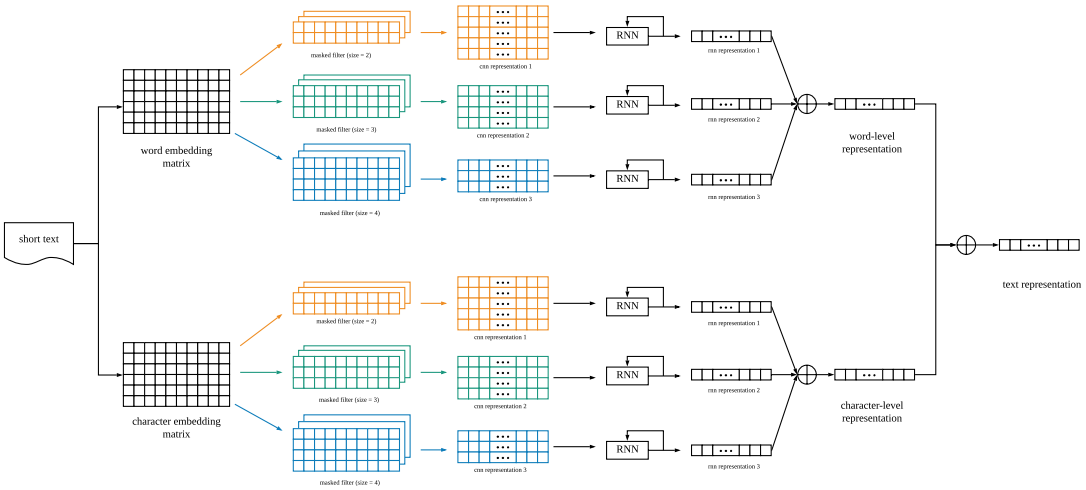


Figure 1. The Architecture of Bi-MACRO.

CNN-RNN network to learn a vector representation. Finally, Bi-MACRO integrates these two vectors to a unified vector as the short text representation.

To tackle the sparsity problem, Bi-MACRO embeds multi-granularity relations among both words and characters by a bi-level multi-scale convolutional neural network (CNN) and recurrent neural network (RNN) structure. To further tolerant noises caused by informal words and typos, Bi-MACRO implements the *breaking-gathering* strategy by a masked CNN layers.

3.2. Bi-level Inputs Transformation

Given a short text $T = \{t_1, t_2, \cdots, t_{n_w}\}$, Bi-MACRO transforms it into a word embedding matrix $\mathbf{E}_w \in \mathcal{R}^{n_w \times n_e^w}$ and a character embedding matrix $\mathbf{E}_c \in \mathcal{R}^{n_c \times n_e^c}$ by looking up the transformation matrices $\mathbf{T}_w \in \mathcal{R}^{nW \times n_e^w}$ and $\mathbf{T}_c \in \mathcal{R}^{nC \times n_e^c}$, where n_w corresponds to the maximum number of words in a short text, n_c corresponds to the maximum number of characters in a short text, n_W refers to the number of unique vocabularies in the corpus, n_C refers to the number of unique characters in the corpus, n_e^w and n_e^c refers to the dimension of the word and character embedding, respectively. Initially, to leverage the word semantic meaning, Bi-MACRO adopts a pre-trained word embedding matrix as $\mathbf{T}_w$ and randomly generates a matrix as $\mathbf{T}_c$. Then, it optimizes $\mathbf{T}_w$ and $\mathbf{T}_c$ in its learning process.

3.3. Masked Convolutional Network

The proposed masked convolutional network is shown in Fig. 2. For a filter, we denote its weight matrix as $\mathbf{W} \in \mathcal{R}^{n_{f_h} \times n_{f_w}}$ where n_{f_h} and n_{f_w} are the height and width of the filter, respectively. Considering the spatial relation of a text matrix is along words and characters instead of embedding features, in Bi-MACRO, we set the filter width n_{f_w} as the dimension of word embedding n_e^w at the word-level and as the dimension of character embedding n_e^c at the character-level. We mask the weight matrix $\mathbf{W}$ by entry-wise multiplying the weight matrix with a mask matrix $\mathbf{M} \in \mathcal{R}^{n_{f_h} \times n_{f_w}}$. Formally, given a word or character embedding matrix $\mathbf{E} \in \mathcal{R}^{n \times n_e}$, we calculate the output of a masked CNN filter (a CNN filter with a masked weight matrix) as

$$\mathbf{o}_{mc} = [o_1, o_2, \cdots, o_{n-n_{f_h}+1}]^\top. \tag{1}$$

In Eq. (1), the k-th entry of $\mathbf{o}_{mc}$ is

$$o_k = g(\sum_{i=1}^{n_{f_w}} \sum_{j=1}^{n_e} \mathbf{M}_{i,j} \mathbf{W}_{i,j} \mathbf{E}_{k+i-1,j} + b), \tag{2}$$

were $g(\cdot) : \mathcal{R} \to \mathcal{R}$ is a non-linear function, and $b \in \mathcal{R}$ is a bias term. In this paper, we use *ReLU* as the non-linear function $g(\cdot)$ in each masked CNN. As demonstrated in Fig. 2, a masked CNN has many filters with the same size to capture different relations among words or characters with the same granularity. Accordingly, for these filters and a set of mask matrices with same entry values, a masked CNN calculates a set of output vectors, and it stacks these vectors as the output matrix:

$$\mathbf{O}_{mc} = [\mathbf{o}_{mc_1}, \mathbf{o}_{mc_2}, \cdots, \mathbf{o}_{mc_{n_f}}]^\top, \tag{3}$$

where n_f refers to the number of filters. The $\mathbf{O}_{mc}$ is also known as the CNN features.

Because the noise may appear at any position between normal words and characters, masks with different mask locations should be adopted. To tackle arbitrary noise positions, masked CNN masks every combination of rows between the first and the last rows for a filter as shown in Fig. 2. Specifically, masked CNN generates $2^{(n_{f_h}-2)}$ different mask matrices for a filter with n_{f_h} height. As a result, it will have $2^{(n_{f_h}-2)}$ CNN feature matrices,

$$O_{mc} = \{\mathbf{O}_{mc}^{(1)}, \mathbf{O}_{mc}^{(2)}, \cdots, \mathbf{O}_{mc}^{2^{(n_{f_h}-2)}}\}. \tag{4}$$

Finally, masked CNN adopts a cross-filter max pooling to integrate its CNN feature matrices as an unified CNN

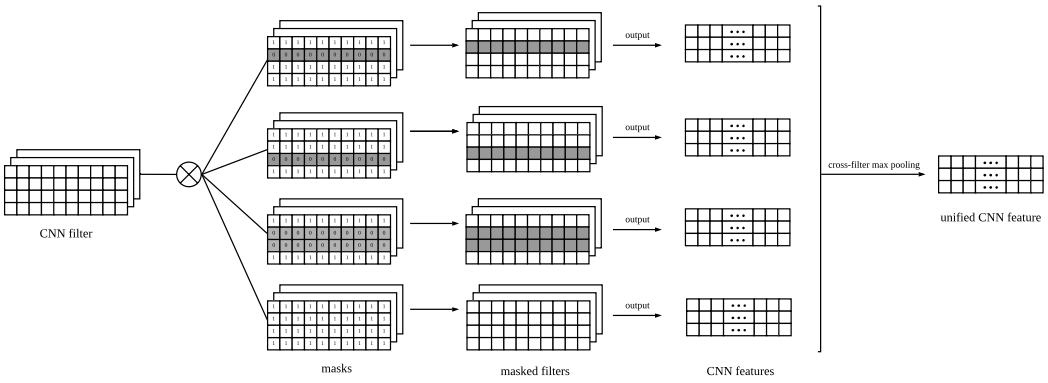


Figure 2. Masked Convolutional Network.

feature matrix. Here, the cross-filter max pooling compares the values at the same entry in different CNN feature matrices and assigns the max value as the value at the same entry in the unified CNN feature matrix. Formally, the value at the (i, j)-th entry in the unified CNN feature matrix can be calculated as:

$$o_{uc_{ij}} = \max(o^{(1)}_{mc_{ij}}, o^{(2)}_{mc_{ij}}, \cdots, o^{2^{(n_{f_h}-2)}}_{mc_{ij}}), \tag{5}$$

where $o^{(k)}_{mc_{ij}}$ is the value at the (i, j)-th entry of the k-th CNN feature matrix in O_{mc}. The rationale is that each entry corresponding to a specific pattern between words or characters, and the largest value corresponding to the most significant pattern. By adopting the cross-filter max pooling, masked CNN can always extract the most significant pattern in text that is tolerant to noise.

3.4. Multi-scale CNN-RNN Structure

In order to capture multi-grain patterns, Bi-MACRO adopts CNN with multi-scale filter sizes. It should be noted that the introduced $2^{(n_{f_h}-2)}$ masks may dramatically increase the model complexity when n_{f_h} is large. Such high model complexity will cause the model learning intractable. To reduce the model complexity, we only use the filter with size $2, 3, 4$, which only increases extra mask matrices by $\frac{4}{3}$ times. With these small size filters, masked CNN is able to capture local noise-tolerant patterns but fail to capture the long-term global relations among words or characters. To fill this gap, Bi-MARCO introduces a recurrent neural network (RNN) after each masked CNN to leverage such long-term global relations. In this paper, we adopt the gated recurrent unit (GRU) to implement the RNN. Specifically, each row in CNN feature matrix O_{mc} is fed into a GRU sequentially. For the t-th row in O_{mc}, the output $\mathbf{h}_t \in \mathcal{R}^{1\times n^r_e}$ of the GRU is computed as follows:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(O_{mc_t}\mathbf{U}^z + \mathbf{h}_{t-1}\mathbf{V}^z),\\
\mathbf{r}_t &= \sigma(O_{mc_t}\mathbf{U}^r + \mathbf{h}_{t-1}\mathbf{V}^r),\\
\hat{\mathbf{h}}_t &= \tanh(O_{mc_t}\mathbf{U}^h + (\mathbf{r}_t \cdot h_{t-1})\mathbf{V}^h),\\
\mathbf{h}_t &= (1-\mathbf{z}_t)\cdot\mathbf{h}_{t-1} + \mathbf{z}_t \cdot \hat{\mathbf{h}}_t,
\end{aligned} \tag{6}$$

where n^r_e is the dimension of the RNN embedding, $\sigma(\cdot) : \mathcal{R}^{1\times n^r_e} \rightarrow \mathcal{R}^{1\times n^r_e}$ is the sigmoid function, $\tanh(\cdot) : \mathcal{R}^{1\times n^r_e} \rightarrow \mathcal{R}^{1\times n^r_e}$ is the tanh function, $\mathbf{r} \in \mathcal{R}^{1\times n^r_e}$ is a reset gate, $\mathbf{z} \in \mathcal{R}^{1\times n^r_e}$ is an update gate, and $\mathbf{U}^z$, $\mathbf{U}^r$, $\mathbf{U}^h$, $\mathbf{V}^z$, $\mathbf{V}^r$, and $\mathbf{V}^h \in \mathcal{R}^{(n_{f_h}-2)^2\times n^r_e}$ are the transform matrices in the GRU. Bi-MACRO uses the last output $\mathbf{h}_{(n_{f_h}-2)^2}$ of the GRU as the RNN output $\mathbf{o}_r$ of a masked CNN.

For masked CNN with filter sizes $2, 3, 4$, Bi-MACRO generates three RNN outputs $\mathbf{o}^{(2)}_r$, $\mathbf{o}^{(3)}_r$ and $\mathbf{o}^{(4)}_r$. Similar to the cross-filter max pooling in the masked CNN, Bi-MACRO here adopts a cross-RNN max pooling to integrate these RNN outputs. Formally, the i-th entry in the unified RNN output $\mathbf{o}_{ur}$ is calculated as:

$$o_{ur_i} = \max(o^{(2)}_{r_i}, o^{(3)}_{r_i}, o^{(4)}_{r_i}). \tag{7}$$

Finally, Bi-MACRO concatenates the unified RNN outputs at the word-level and character-level to form the short text representation:

$$\mathbf{o} = [\mathbf{o}^{w\top}_r, \mathbf{o}^{c\top}_r]^\top. \tag{8}$$

The short text representation $\mathbf{o}$ is then fed into a downstream text analytic tasks such as text category classification and sentiment classification. The Bi-MACRO is jointly trained with the downstream task to represent short text in an end-to-end fashion.

4. Experiments

4.1. Experiment Setup

We compare Bi-MACRO with three state-of-the-art short text representation methods, including WWE [10], SIF [11], and SeaNMF [22], and two baseline methods, including tf-idf and LDA, to evaluate Bi-MACRO's performance.

We conduct experiments on five widely used real-word short text data sets. They include question answering data sets: TREC[1], Quora[2]; social media data sets: Twitter[3]; article

1. http://cogcomp.cs.illinois.edu/Data/QA/QC
2. https://www.kaggle.com/c/quora-question-pairs/data
3. https://www.cs.york.ac.uk/semeval-2013/task2.html

TABLE 1. THE DATA CHARACTERISTICS OF EACH DATA SET.

Data Set	TREC	Quora	Twitter	News	AG News
#Texts	10,764	537,933	11,394	20,120	471,542
#Class	6	2	3	8	5
#Voc.	8,872	105,929	20,587	24,201	70,592
#Unk Voc.	3,774	48,699	13,782	7,936	51,773
Avg. Len.	6.10	12.97	12.53	9.44	4.53
Avg. Char.	29.09	61.86	59.98	72.39	21.28

title data sets: News[4], AG News[5]. For AG News data set, we select five comparable categories: entertainment, sports, business, sci/tech, and health. For AG News and Quora data sets, we randomly use 95% and 5% objects in the data set as the training and testing data, respectively. For other data sets, we use the original provided training and testing sets. The characteristics of each data set are shown in Tab. 1. As shown in the Tab. 1, the noise and sparsity are appeared in all data sets. Data set with larger ratio of unknown words represents more noise, and that with shorter length shows a larger degree of sparsity.

For Bi-MACRO, we empirically set the number of filters with the same size as 100, the number of RNN units as 100, the word-level and character-level text embedding dimension as 100. We use *ReLU* function as the activation function in each hidden unit in Bi-MACRO, and adopt *Adam* [23] as the optimization method to train the Bi-MACRO with batch size 32. For the competitors, we adopt their default setting. We use the pre-trained word embedding by GloVe algorithm [24] in Bi-MACRO, WWE, and SIF. For unknown words, we randomly initialize their embeddings.

4.2. Effectiveness on Short Text Classification

For all data sets instead of Quora, we feed the representation of each method into a three-layers fully-connected neural network to classify short texts. We set the number of hidden units in each hidden layer as 100 and use *tanh* as the activation function in each hidden unit. The label of Quora data set is whether two sentences are the same question, i.e. with the same meaning. Accordingly, we first adopt each representation methods to represent the sentences, and then, concatenate the representations of two sentences as the input of a classifier to classify whether they are the same question. Here, we also use a three-layers fully-connected neural network with the same setting above as the classifier.

We report the accuracy of the short text classification enabled by different methods in Tab. 2. As can be seen in Tab. 2, Bi-MACRO enables the best performance in all data sets. We further show the performance improvement ratio (Δ) of Bi-MACRO compared with the other method with the highest accuracy. Bi-MACRO improves up to 4.87% on AG News data set, which has the most significant noise and sparsity as shown in Tab. 1. It not only illustrates Bi-MACRO significantly improves the short text classification performance but also demonstrates Bi-MACRO effectively tackles the noise and sparsity in short text representation.

4. http://acube.di.unipi.it/tmn-dataset/

5. http://www.di.unipi.it/ gulli/AG_corpus_of_news_articles.html

TABLE 2. THE SHORT TEXT CLASSIFICATION PERFORMANCE BASED ON DIFFERENT REPRESENTATION METHODS.

Data Set	TREC	Quora	Twitter	News	AG News
Tf-idf	96.80	78.62	59.67	63.90	71.44
LDA	77.20	74.17	49.70	43.86	44.98
WWE	96.80	78.94	44.96	15.44	61.11
SeaNMF	25.40	61.87	48.75	16.84	22.11
SIF	95.80	77.98	60.26	77.89	75.37
Bi-MACRO	**98.00**	**81.61**	**61.45**	**79.56**	**79.04**
Δ	1.24%	3.38%	1.97%	2.14%	4.87%

TABLE 3. THE SHORT TEXT INFORMATION RETRIEVAL PERFORMANCE BASED ON DIFFERENT REPRESENTATION METHODS.

Metric	Precision@5	Precision@10	Precision@20
Tf-idf	42.65	40.50	38.31
LDA	34.23	32.51	31.07
WWE	45.49	44.12	42.70
SeaNMF	44.36	41.95	40.69
SIF	55.04	52.83	50.48
Bi-MACRO	**70.76**	**70.56**	**70.34**
Δ	28.56%	33.56%	39.34%

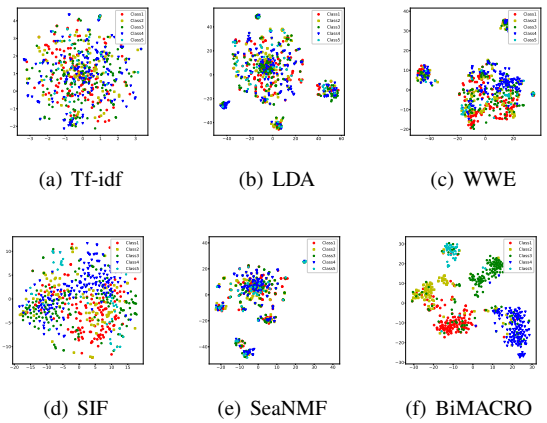
4.3. Effectiveness on Short Text Retrieval

We further test the Bi-MACRO representation performance by short text retrieval. The short texts in the testing set are used as queries, and precision@k, i.e., the fraction of k-closest short texts selected per the Euclidean distance in a representation space that are the same-class neighbors, is reported. We conduct the short text retrieval experiments on the AG News data set because it has the most significant noise and sparsity. Different from the short text classification, in which performance may be also affected by the classifier, the short text retrieval performance only depends on the text representation itself. Besides, the short text classification evaluates the short text representation from global distribution perspective, while the short text retrieval can inspect it from local perspective, especially when k is small. Thus, we select k as 5, 10 and 20 in precision@k.

We report the results of short text retrieval in term of precision@k in Tab. 3. Bi-MACRO significantly improves the performance of short text retrieval (up to 39.34%). In addition, with the number of retrieval texts (k) increasing, the precision@k of all methods instead of Bi-MACRO decreases rapidly. It reflects Bi-MACRO captures much more noise-tolerant patterns in multiple granularity, which preserves the short text local distribution compared with other methods.

4.4. Quality of Short Text Representation

We visualize the short text representation on AG News testing data set in a two-dimensional space trough TSNE [25]. To evaluate the representation quality, we plot the



(a) Tf-idf (b) LDA (c) WWE

(d) SIF (e) SeaNMF (f) BiMACRO

Figure 3. Short Text Representation of Different Methods.

category label of each short text by different colors. A high-quality short text representation will clearly separate texts in different categories in the representation space.

In the Bi-MACRO generated representation space, the short texts in the same category are clearly clustered together. In contrast, the representation of other methods mixes the texts with different categories. This is because Bi-MACRO captures the multi-grain noise-tolerant patterns in short text by the bi-level masked multi-scale CNN-RNN structure, which significantly filters the noise and fits the sparse text. As a result, the semantic meaning of a short text is propriety reflected by Bi-MACRO's representation.

5. Conclusion

This paper proposes a bi-level masked multi-scale CNN-RNN networks to tackle the noise and sparsity problems in short text representation. The proposed representation method learns multi-grain noise-tolerant patterns and then embeds the most significant patterns in a short text as its representation. It can effectively represents short text and significantly improves the downstream analytic tasks as demonstrated by comprehensive experiments.

References

[1] K. Dey, R. Shrivastava, and S. Kaushik, "A paraphrase and semantic similarity detection system for user generated short-text content on microblogs," in *COLING*, vol. 42, 2016, pp. 2880–2890.

[2] H. Amiri, H., & Daumé III, "Short text representation for detecting churn in microblogs," in *AAAI*, 2016, pp. 2566–2572.

[3] Z. Ji, F. Xu, B. Wang, and B. He, "Question-answer topic model for question retrieval in community question answering," in *CIKM*. ACM, 2012, pp. 2471–2474.

[4] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei, "Author topic model-based collaborative filtering for personalized poi recommendations," *IEEE transactions on multimedia*, vol. 17, no. 6, pp. 907–918, 2015.

[5] D. Zhang, Y. Li, J. Fan, L. Gao, F. Shen, and H. T. Shen, "Processing long queries against short text: Top-k advertisement matching in news stream applications," *ACM Transactions on Information Systems (TOIS)*, vol. 35, no. 3, p. 28, 2017.

[6] K. Dey, R. Shrivastava, and S. Kaushik, "A paraphrase and semantic similarity detection system for user generated short-text content on microblogs," in *COLING*, 2016, pp. 2880–2890.

[7] J. Wang, Z. Wang, D. Zhang, and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification," in *IJCAI*, vol. 350, 2017.

[8] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[9] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *EMNLP*, 2015, pp. 1422–1432.

[10] C. De Boom, S. Van Canneyt, T. Demeester, and B. Dhoedt, "Representation learning for very short texts using weighted word embedding aggregation," *Pattern Recognition Letters*, vol. 80, pp. 150–156, 2016.

[11] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *International Conference on Learning Representations*, 2017.

[12] X. Li, Y. Wang, A. Zhang, C. Li, J. Chi, and J. Ouyang, "Filtering out the noise in short text topic modeling," *Information Sciences*, vol. 456, pp. 83–96, 2018.

[13] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, and H. Xiong, "Topic modeling of short texts: A pseudo-document view," in *SIGKDD*. ACM, 2016, pp. 2105–2114.

[14] S. Liang, E. Yilmaz, and E. Kanoulas, "Dynamic clustering of streaming short documents," in *SIGKDD*. ACM, 2016, pp. 995–1004.

[15] J. V. Lochter, R. F. Zanetti, D. Reller, and T. A. Almeida, "Short text opinion detection using ensemble of classifiers and semantic indexing," *Expert Systems with Applications*, vol. 62, pp. 243–249, 2016.

[16] X. Li, C. Li, J. Chi, and J. Ouyang, "Short text topic modeling by exploring original documents," *Knowledge and Information Systems*, vol. 56, no. 2, pp. 443–462, 2018.

[17] C. T. Zheng, C. Liu, and H. San Wong, "Corpus-based topic diffusion for short text clustering," *Neurocomputing*, vol. 275, pp. 2444–2458, 2018.

[18] E. Kontopoulos, C. Berberidis, T. Dergiades, and N. Bassiliades, "Ontology-based sentiment analysis of twitter posts," *Expert systems with applications*, vol. 40, no. 10, pp. 4065–4074, 2013.

[19] M. M. Mostafa, "More than words: Social networks text mining for consumer brand sentiments," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4241–4251, 2013.

[20] V. Nastase and M. Strube, "Transforming wikipedia into a large scale multilingual concept network," *Artificial Intelligence*, vol. 194, pp. 62–85, 2013.

[21] H. Lu, L.-Y. Xie, N. Kang, C.-J. Wang, and J.-Y. Xie, "Don't forget the quantifiable relationship between words: Using recurrent neural network for short text topic discovery." in *AAAI*, 2017, pp. 1192–1198.

[22] T. Shi, K. Kang, J. Choo, and C. K. Reddy, "Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations," in *WWW*, 2018, pp. 1105–1114.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.

[25] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.