2021 4th International Conference on Artificial Intelligence and Big Data

A Novel Microblog Sentiment Classification Method Based on Top-k Pooling

Binyan Zhang
Computer Science and Engineering
Chongqing University of Technology
Chongqing, China
zby@2018.cqut.edu.cn

Xiaofei Zhu*
Computer Science and Engineering
Chongqing University of Technology
Chongqing, China
zxf@cqut.edu.cn

Xianying Huang
Computer Science and Engineering
Chongqing University of Technology
Chongqing, China
hxy@cqut.edu.cn

Wanping Liu
Computer Science and Engineering
Chongqing University of Technology
Chongqing, China
wpliu@cqut.edu.cn

***Abstract*—Microblog is a popular social media platform for information sharing and dissemination. Recently, microblog sentiment classification has received a lot of attention in many real-world applications, such as stock price prediction, public opinion monitoring, and crisis management. Existing microblog sentiment classification methods identify the sentiment polarity mainly based on the textual content. As microblog is usually short and contains a lot of noisy, it is very challenge to learn powerful representation only relied on the textual information. In this paper, we argue that the multi-head self-attention, which can capture complicated interactions information between words in sequence, would inevitably to introduce lots of noisy information. Moreover, the user sentimental polarity also plays an important role in judging the sentiment of text. Therefore, we propose a novel microblog sentiment classification method, called Multi-head Self-attention Based on Top-k pooling (MSBT). Specifically, we design a Top-k pooling layer to alleviate the issue caused by the multi-head self-attention network, and then incorporate user historical sentiment tendency in the loop of microblog sentiment classification. Extensive experiments conducted on a real-word dataset MDUHI demonstrates that our proposed approach MSBT can significantly improve the performance of microblog sentiment classification, e.g., the F1 score is 0.98% higher than the optimal benchmark method NPA, reaching 93.39%.**

Keywords—sentiment classification, multi-head self-attention, Top-k pooling, microblog analysis

I. Introduction

Microblogging services have attracted a large number of users to participate in information sharing and dissemination. With the massive growth of microblogs posted by users, how to effectively identify users' sentiments has become a crucial research problem and played a key role in many real-world applications. Microblog Sentiment classification, which aims to classify microblogs into positive, negative or neural based on the text, is a fundamental task for sentiment analysis.

Recently, many studies propose to take the relations among microblogs into consideration [1]. Yang [2] assume that microblogs are networked data and the social context information should be considered. To the end, they design a deep learning method with attention mechanism to fully capture the features of microblog relations in order to promote microblog sentiment classification results. Zou [1] introduce structure similarity context into social contexts, and further leverage the semantic relations between microblogs. The major shortcomings of these methods are that they mainly rely on the social contexts and would be impractical when such information is not available.

In this paper, we argue that in addition to the textual content or social context, user historical sentimental polarity also has a critical influence on judging the sentiment of her text. For example, a user with positive attitude towards life would tend to post positive texts on social media. Furthermore, we also argue that utilizing multi-head self-attention to capture complicated words interactions information in sequence would results in that the output vectors contains loss of irrelevant signals.

Based on the above mentioned observations, we propose a novel microblog sentiment classification method, called MSBT. In particular, we propose to enhance the word representation by introducing word sentimental information into its representation. A BiLSTM is utilized to capture the rich contextual information of words in the sequence. In order to capture the complicated interaction information between words, we employ the multi-head self-attention network to model information from different representation sub-spaces. To handle the issue of the multi-head self-attention network, we design a Top-k pooling layer to highlight important elements. After that, we further involve user historical sentiment tendency in the loop of identifying the sentimental polarity of her posted text sequences. Experimental results on the benchmark dataset MDUHI show that our proposed method can significantly improve the performance of the microblog sentiment classification. The main contributions of this paper are as follows:

1) We employ the multi-head self-attention associated with a Top-k pooling module. As the multi-head self-attention can capture complicated interactions information between words in sequence, and it will inevitably to introduce lots of noisy information. The Top-k pooling module is designed to alleviate the noise issue by discarding irrelevant signals.

This research was funded by the National Natural Science Foundation grant number 61722211 and Chongqing University of Technology grant number clgycx20203113.

978-0-7381-3169-6/21/$31.00 ©2021 IEEE

2) We incorporate user historical sentiment tendency in the loop of identifying the sentimental polarity of her posted text sequences.

3) We conducted extensive experiments and results show that our proposed approach MSBT can significantly improve the performance of microblog sentiment classification.

II. Related work

Sentiment dictionary has been widely used in sentiment classification of microblog and achieved encouraging performance. For example, Hu [3] brought a group of manually collected and annotated sentiment dictionaries, and searched for their near-antonyms and incorporated them into the sentiment dictionaries to form the final sentiment dictionaries. Kennedy [4] not only considered the influence of positive and negative words, but also account for valence shifters which can change the semantic orientation of another word. Taboada [5] suggesting that corresponding weights should be added to intensification and negation. Due to the problem that the sentiment classifier usually tends to make positive prediction, they increased the weight of all negative emotion words by 50% to solve this problem.

Machine learning methods usually rely on extracting effective features of text. Pang [6] combined SVM, ME and NB based on N-gram features to analyze the sentimental orientation of the movie review. Mullen [7] combined PMI, POS, and sentiment value of sentiment words, with N-gram to obtain enhanced features and improved the classification effect considerably. Recently, deep learning has achieved great success in the task of sentiment classification. Kim [8] first applied the trained word vectors to a CNN to predict sentence-level textual sentiment. Dong [9] trained the sentence vector of text through the RNN, and constructed the text feature vector from the sentence vector. They improved the accuracy of text sentiment classification considerably. Yang [10] utilized a hierarchical attention network for document classification, which is based on the hierarchical structure of documents as well as the attention mechanism to word and document representation which are extracted by the BiLSTM. More recently, Vaswani [11] proposed multi-head self-attention, which makes every word have global semantic information by paying attention to words in a sentence, and used different heads to learn different subspace semantics.

Although these methods have achieved satisfactory performance, they still face the following limitations. First, these models mainly model the semantic information and structural information of the text, while ignore the influence of user historical sentiment tendency on microblog sentiment classification. Second, the multi-head self-attention mechanism can capture complicated words interaction information in sequence, however, it would also cause in that the output vectors contains loss of irrelevant signals. Our work differs with existing methods in two folds: (1) we introduce a novel Top-K pooling module on top of the multi-head self-attention module, which is designed to remove noisy elements; (2) we incorporate user historical sentiment tendency into our proposed model in order to guide the sentiment classification.

III. Method

In this section, we introduce our proposed approach, named MSBT. Our approach consists of four modules: 1) Sentiment-aware word representation module, which aims to capture the semantic as well as sentimental information of the input text; 2) Context-aware word representation module, which obtains more informative representations of microblog by modeling the contextual information of words within the input text; 3) Top-k pooling module, which is used to extract the most important k information units from each word representations and obtains the representation of the input text based on a concatenation operation; 4) User historical sentiment module, which integrates user historical sentiment tendency into the learned representation of the Top-k pooling module and outputs the final classification results with a softmax layer. Fig. 1 shows the framework of MSBT.

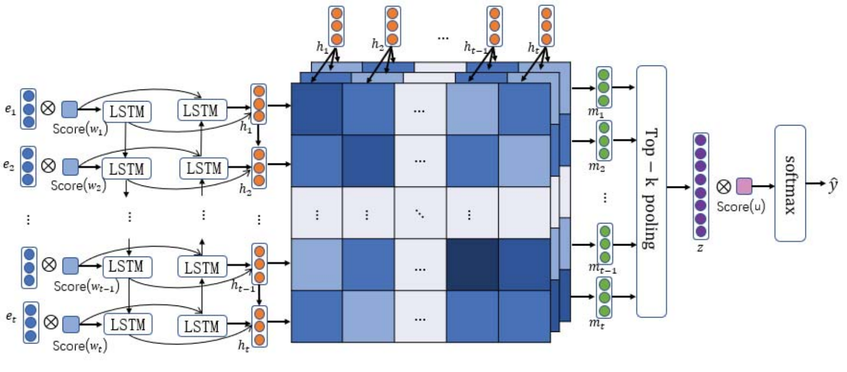

Fig. 1. The framework of MSBT.

A. Sentiment-Aware Word Representation

In this module, we build the representation of words in the input text. To obtain words' semantic information, we leverage the Wikipedia word vector trained by Word2Vec to learn the embedding for each word, which contains a vector representation of 575,746 words in 200 dimensions.

In addition, we employ sentiment dictionary to simultaneously model the semantic information and sentimental information of the text. When constructing the sentiment dictionary, we use HOWNET as the base dictionary.

Due to the existence of a large number of network language in microblog, we conduct manual sentiment annotations on the commonly used words, sentimental symbols and sentimental emoticons, and merge the marked sentimental words with HOWNET sentimental dictionary, and finally we get 4182 sentimental words. Then the sentimental score is calculated based on the frequency of sentimental words in different polarity sequence sets.

We use formula (1) to calculate the frequency of sentimental words in sequences of different polarities $F(w_i)$:

$$F(w_i) = |\alpha \cdot P(w_i) - \beta \cdot N(w_i)| \tag{1}$$

while $P(w_i)$ and $N(w_i)$ represent the frequency of the i-th sentiment word w_i in positive and negative sequences, respectively. α and β represent the importance parameters of $P(w_i)$ and $N(w_i)$. The sentiment score of w_i is $Score(w_i)$, which is calculated as follows:

$$Score(w_i) = \left[\frac{F(w_i) - F_{min}}{F_{max} - F_{min}} \cdot \gamma\right] \tag{2}$$

where F_{min} and F_{max} represent the minimum sequence frequency and maximum sequence frequency of sentiment word appearing in the dataset, respectively. $|\cdot|$ is the operation of taking absolute values, $[\cdot]$ for rounding operation. γ is a threshold for controlling the score of sentiment words. For other words that don't appear in the sentiment dictionary, their sentiment scores are computed as follows:

$$Score(\widehat{w}_i) = [\alpha \cdot P(\widehat{w}_i) - \beta \cdot N(\widehat{w}_i)] \tag{3}$$

where $P(\widehat{w}_i)$ and $N(\widehat{w}_i)$ represent the frequency of the i-th non-sentiment word in positive and negative sequences, respectively.

For a microblog sequence $s = \{w_1, w_2, \cdots, w_L\}$, we embed the text sequence into an embedding sequence $E = \{e_1, e_2, \cdots, e_L\}$ using the embedding matrix $W^e \in R^{d \times N}$ as we discussed above. Formally, we have:

$$e_j = W^e v_j, \ 1 \leq j \leq t \tag{4}$$

where N denotes the number of words in the dictionary, and d represents the word vector dimension. $v_j \in \{0,1\}^N$ represents a one-hot vector corresponding to word w_j. Then, we combine the word w_j's embedding e_j and its corresponding sentiment word score to obtain the final word representation, i.e., r_j, which will be used as the input for the next layer.

$$r_j = \begin{cases} e_j \otimes Score(w_i), w_i \in Sentiment\ words \\ e_j \otimes Score(\widehat{w}_i), \widehat{w}_i \in Non-sentiment\ words \end{cases} \tag{5}$$

where $\otimes$ denotes the element-wise multiplication.

B. Context-Aware Word Representation

This module aims to capture the contextual information of words in the microblog sequence, which has two sub-modules, which are described as follows: The first one is a BiLSTM network. As the remote information is very important to identify the sentiment classification of a sequence. BiLSTM is composed of both forward and backward networks, which is used to model the context dependence of words. The word hidden vector r_j obtained from the first module only independent information of word $\mathrm{w_j}$ in the sequence. We utilize BiLSTM to capture the contextual information of each word $\mathrm{w_j}$.

$$\overrightarrow{h_j} = \overrightarrow{LSTM}(r_j, \overrightarrow{h_{j-1}}) \tag{6}$$

$$\overleftarrow{h_j} = \overleftarrow{LSTM}(r_j, \overleftarrow{h_{j+1}}) \tag{7}$$

where $\overrightarrow{h_j} \in R^u$ and $\overleftarrow{h_j} \in R^u$ denote the hidden states of the forward LSTM and backward LSTM, respectively. Then we concatenate $\overrightarrow{h_j}$ and $\overleftarrow{h_j}$ to form the hidden representation h_j, i.e., $h_j = [\overrightarrow{h_j}; \overleftarrow{h_j}]$. We represent all hidden output as $H \in R^{t \times 2u}$:

$$H = (h_1, h_2, \ldots, h_L) \tag{8}$$

The second is the multi-head self-attention network. As self-attention is used to capture the interaction information between words in sequence, a word can interact with more than one word, so we propose to employ the multi-head self-attention mechanism to build a textual representation of a high-quality word by jointly plotting the interaction between the word and more than one word. Multi-head self-attention allows the model to focus on information from different representation sub-spaces. In the multi-head self-attention network, the representation vector $m_{i,j}$ of the j-th word generated by the i-th self-attention head is obtained by weighted summation of all word vectors in the output sequence H of the upper layer. The calculation process is shown in formulated as follows:

$$\hat{\alpha}^i_{j,k} = h_j^T U_i h_k \tag{9}$$

$$\alpha^i_{j,k} = \frac{exp(\hat{\alpha}^i_{j,k})}{\sum_{m=1}^{t} exp(\hat{\alpha}^i_{j,m})} \tag{10}$$

$$m_{i,j} = W_i\left(\sum_{m=1}^{t} \alpha^i_{j,m} h_m\right) \tag{11}$$

In the formula, U_i and W_i are the projection parameters of the i-th self-attention head, and $\alpha^i_{j,k}$ represents the relative importance of the interaction between the j-th word and the k-th word.

By modeling the interaction between a word and all words in the text, we can get the hidden layer representation of each word, and each hidden layer representation is constructed by the hidden representation of all words in the text. The multi-head representation of the j-th word m_j is to concatenate all the representation vectors generated by h independent self-attention headers, as shown in formula (12):

$$m_j = [m_{1,j}; m_{2,j}; \cdots; m_{m,j}] \tag{12}$$

C. Top-k Pooling

In the text matching task, several words in the sequence that are more relevant to the query will determine the final score, while irrelevant signals, may have a negative impact. Therefore, selecting the k largest elements is beneficial to remove noise and improve matching accuracy. We propose a Top-k pooling layer to select the Top-k elements from each word vector, which will be concatenated as the final representation of the sequence. The model structure of the Top-k pooling layer is shown in Fig. 2.

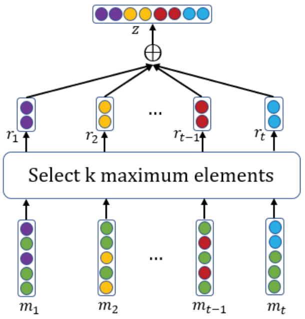

Fig. 2. The framework of Top-k pooling Layer.

The rationality of applying the Top-k pooling is that: Compared with the max pooling, which only keep the maximum element of a word vector and would ignore a lot of sub-relevant signals. Compare with the average pooling, which considers all elements of a word vector, may introduce lots of noisy information. Therefore, the Top-k pooling can be considered as a trade-off between the max pooling and the average pooling. After the operation is completed, a fixed value will be obtained to provide conditions for entering the fully connected neural network, as shown in formula (13):

$$\tilde{z} = Concat\left(Topk(\mathrm{m_j}), j \in (1, 2, \ldots, \mathrm{L})\right) \tag{13}$$

where $Topk(\cdot)$ is the Top-k pooling function, $Concat(\cdot)$ is the concatenation operation, and $\tilde{z}$ is the output of the Top-k pooling layer.

D. User Historical Sentiment Module

As user historical sentiment tendency usually have a strong influence to the potential sentiment of her posted microblogs, it is beneficial to further involve user historical sentiment tendency in the loop of identifying the sentimental polarity of her posted text sequences.

Suppose u is the corresponding user of the input text s, and $F(p)$ and $F(n)$ are the number of positive texts and negative texts posted by user u in a period of time, respectively. $F(N_i)$ is the amount of non-emotional text posted by user u during this period. Then, we define u 's positive sentimental orientation score $Score(u^p)$ and her negative sentimental orientation score $Score(u^n)$ as follows:

$$Score(u^p) = \frac{F(p)}{F(p) + F(n) + log\left(F(n)\right)} \tag{14}$$

$$Score(u^n) = \frac{F(n)}{F(p) + F(n) + log\left(F(n)\right)} \tag{15}$$

Then the overall sentimental orientation score $Score(u)$ of user u is formalized as:

$$Score(u) = \begin{cases} \mu \times Score(u^p), & Score(u^p) \geq Score(u^n) \\ \frac{\mu}{2} \times Score(u^n), & Score(u^p) < Score(u^n) \end{cases} \tag{16}$$

where μ is an adjustable parameter for adjusting the importance of scores of different polarities, mapping the user's positive sentimental tendency score to the interval [0.5,1], and mapping the user's negative sentimental tendency score to the interval [0,0.5), which is used to facilitate the feature Matrix calculation. Finally, the output obtained by the Top-k pooling layer and the corresponding user's overall sentimental tendency score are merged as follows:

$$z = \tilde{z} \otimes Score(u) \tag{17}$$

where $\otimes$ denotes the multiplication operation.

E. Model Training

We use a fully connected layer with ReLU to convert the learnt hidden representation z, which is shown in formula (18):

$$z' = ReLU(W \times z + b) \tag{18}$$

then we use a softmax layer to output the prediction $\hat{y}$ as follows: (19):

$$\hat{y} = softmax\left({W'}^T \times z' + b'\right) \tag{19}$$

where W, W' , b' and b are trainable parameters.

The loss function ℓ used in this paper is shown in formula (20):

$$\ell = -\frac{1}{|S|} \sum_{y \in S} \sum_{k=1}^{2} y_k \, log(\hat{y}_k) \tag{20}$$

where S represents the dataset, y_k represents the true label.

IV. Experiment

In this section, we empirically evaluate the proposed MSBT model on a real-world microblog dataset, and the experimental results demonstrate the effectiveness of our proposed model.

A. Dataset

We create a new Microblog Dataset with User Historical Information (MDUHI). In order to focus on general and active users, we randomly selected 200 users who have 50-50000 followers and 100-1000 posted microblogs. We crawled their microblogs and then randomly selected 10,000 microblogs for manual annotation. After annotation, we only keep microblogs with clear sentimental polarity and finally obtain 3,000 entries.

In addition, we further process the data by removing stop words. In our work, we use a stop word table from Harbin Institute of Technology which contains 1893 stop words and useless symbols, such as: "!" , ";" , "and".

B. Settings

We set the frequency of documents importance parameters α and β to 0.3 and 0.4, respectively. For model training, we use Adam, and sets the learning rate to 0.01. The classification performance of the model is also affected by different weights of users' feature μ. We select μ as 0.8 and given more discussion about the influence of μ in Parameter Sensitivity.

Table 1 presents the important parameter settings of MSBT.

TABLE I. Model Parameter Setting Table

Parameter	Value
Word Vector Dimension	200
User Feature Weight μ	0.8
Regular Weight Restrictions	2

Dropout	0.9
Number of Heads h	16
Value of k	3

C. Baselines

We compare our proposed approaches with several competitive deep leaning based sentiment classification methods. Details of the baselines are as follows:

- SD [12]: SD first extends the sentiment dictionary by extraction and construction of degree adverb dictionary. Then it estimates the sentiment value of a text via the calculation of the weight.
- SVM [13]: This method combines semantic information in both word2vec and tf-idf. It assumes that the word2vec offers extra semantic features that can't captured by tf-idf.
- W2V + CNN [14]: This method is a model based on deep learning. It first trains word vectors using Word2Vec, and then employs a CNN to learn representation of text for sentiment classification.
- TNA [14]: It proposes a novel two channels neural network

  structure with attention mechanism. It uses a LSTM to capture semantic information, and a Tree structure LSTM channel to obtain syntactic information. Then, a sentence-level attention mechanism for word sequences is leveraged to determine most influential component.
- EV-CNN [15]: It uses wiki Chinese dataset and network terms to expand the original vocabulary, train new word embedding, and implements sentence level sentiment classification based on convolution neural network. An optimization method based on pooling level sentence length is proposed.
- HAN [10]: HAN proposes a hierarchical attention network for document classification, which uses a two levels of attention mechanism at word level and sentence level respectively. It relies on two key signals: 1) documents usually have a hierarchical structure, such as words of sentence, sentences of document; 2) different words and sentences in a document would be deferentially informative.
- NPA [16]: This method proposes to apply the attention mechanism at both the word level and the news level to help the model pay attention to important words and news. It finally embeds the user's sentimental tendency information implicitly by employing the attention mechanism to generate query vectors for text and news-level attention

D. Overall Performance

The evaluation metrics used in this paper are : precision, recall and F1, which are the three most commonly used metrics in natural language processing. Table. 2 shows the evaluation results of different models on MDUHI.

TABLE II. TEST RESULTS OF DIFFERENT MODELS ON THREE INDICATORS (P (%) , R (%) , F1 (%))PARAMETER SETTING TABLE

Model	P_1	P_0	R_1	R_0	F1_1	F1_0	P_Ave	R_Ave	F1_Avg
SD	77.0	68.0	42.0	91.0	54.0	78.0	71.0	70.0	70.0
SVM	76.0	80.0	69.0	85.0	72.0	82.0	78.0	78.0	78.0
W2V+CNN	85.0	81.0	74.0	90.0	79.0	85.0	83.0	83.0	83.0
TNA	91.0	80.0	72.0	94.0	80.0	86.0	85.0	84.0	84.0
EV-CNN	91.0	86.0	79.0	95.0	84.0	90.0	88.0	88.0	88.0
HAN	88.97	96.55	97.78	83.69	93.16	89.65	92.22	91.77	91.67
NPA	90.22	96.53	93.81	85.13	93.81	90.43	92.84	92.5	92.41
MSBT	**90.95**	**97.81**	**98.62**	**86.82**	**94.63**	**91.75**	**93.82**	**93.48**	**93.39**

We observe from Table 2, MSBT performs the best and significantly outperforms all baseline models on MDUHI. The SD is a traditional microblog sentiment classification method, and it obtains the worst F1 70%. The SVM method demonstrates a better performance than the SD method, reaching F1 78%. Because the SVM method can effectively model the nonlinear data. The W2V+CNN based on the convolutional neural network model is 6.4% better than the SVM. This is due to the superior modeling capability of deep learning. The classification effect of TNA is higher than W2V+CNN, and the score of F1 reaches 84%. As this method takes advantage of the attention mechanism and the feature of Tree-LSTM that can effectively exploit the structural features of statements. EV-CNN models the sentiment score and weight score of words, adds sentiment information to the model to help improve the classification performance. HAN applied the attention mechanism of two levels to word level and document level respectively to jointly mine the information of words and documents. In this paper, we apply the attention mechanism in our model, which selects word items with qualitative information, and achieves a F1 of 91.67%. NPA proposes a mechanism to embed users' sentimental orientation information into word and news-level attention mechanism is utilized to help models focus on important words and news. Finally, the F1 of NPA reaches 92.41%, because NPA not only focuses on extracting the text context information, but also models the user's sentimental tendency information.

E. Parameters Sensitivity

- Parameter μ

The parameter μ reflects the importance of users' feature in our method. To analyze the influence of μ, we fix the parameters μ to 0.8, and vary μ from 0 to 1 with a step size 0.1. From Fig. 3, we observe that the performance raises first and reaches a peak when $\mu = 0.8$. After that the performance drops slowly.

- Parameter h

The parameter h controls the influence of the number of heads in multi-head self-attention module. Fig.3 demonstrates the impact of h to MSBT on MDUHI dataset. To be specific, we vary h as 1, 2, 4, 8, 16, 32, while fixing the other parameters to the following values: μ =0.8 and $k = 3$. We can observe that the performance increases as we increase h, and reaches the best performance when h is 16, which is followed by a considerable drop of performance. This is because when $h = 1$, it equals to the self-attention model.

- Parameter k

The value of k determines the number of most important element we choose. Fig.3 shows the performance of our proposed model MSBT when we vary the parameter k from 1 to 6 with a step size 1. We can observe that the performance raises first and reaches the peak when $k = 3$. Then it starts to drop slowly. Generally, k is stable when k is within the range from 3 to 6. The reason is that when the value of k is small, a lot of useful information will be ignored. When the value of k becomes too large, some noisy element would be considered and results in sub-optimal performance.

F. Embedding Visualization

In order to show the superior embedding quality of MSBT over other methods, we use t-SNE [17] to visualize the learned representations of microblogs for comparison. Fig. 4 shows the visualization results of our proposed approach MSBT, as well as two best performing baselines, i.e., HAN and NPA on the test microblog of MDUHI. The two node colors indicate with different labels are mixed together. For example, in Fig. 4 (a), several nodes with red color are mixed with the main cluster of green class, while in Fig. 4 (b), a small set of nodes with green color are separated from the main cluster of green class. Compared with both HAN and NPA, our method demonstrates a clear separation of nodes with different class labels This is because the two baselines only consider extracting rich feature information, while neglect the noisy issue. Our method alleviate this issue by introducing a novel Top-k pooling layer which selects salient elements within the representation and discards less important as well as noisy elements.

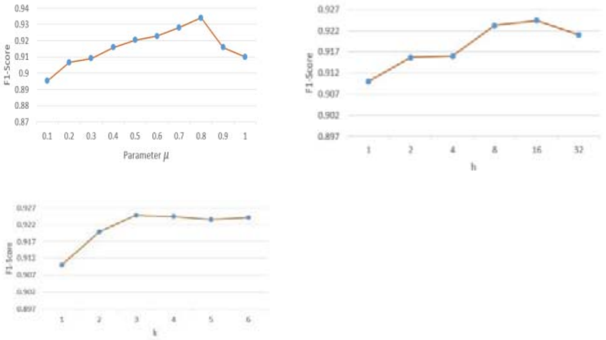

Fig. 3. F1 with different μ, h and k on MDUHI.

V. Conclusion

As in this work we solely consider the user historical sentiment tendency in a coarse-grained mode which would neglect a lot of useful information. In the future, we will investigate how to capture user rich historical sentiment information with a fine-grained model.

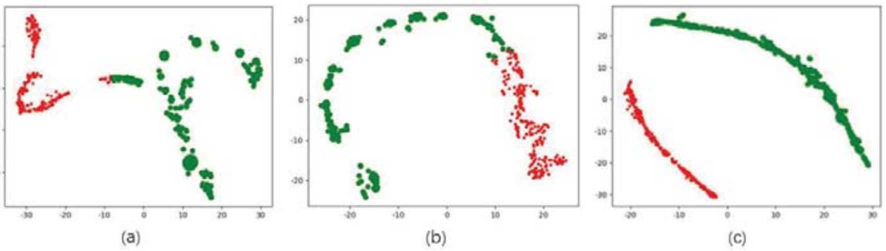

Fig. 4. t-SNE visualization comparison of the embedding feature space learned by three different methods HAN, NPA, MSBT.

References

[1] Xiaomei, Zou; et al. ConSent: Context-based sentiment analysis. Plos One.13(2),e0191163(2018).

[2] Yang, Jing; et al. Microblog sentiment analysis via embedding social contexts into an attentive LSTM. Engineering Applications of Artificial Intelligence.97,104048 (2021).

[3] Hu, Minqing; et al., Mining and summarizing customer reviews. Tenth Acm Sigkdd International Conference on Knowledge Discovery Data Mining. ,(2004).

[4] Kennedy, et al. SENTIMENT CLASSIFICATION of MOVIE REVIEWS USING CONTEXTUAL VALENCE SHIFTERS. Computational Intelligence. 22(2) ,110-125(2010).

[5] Taboada, et al. Lexicon-based methods for sentiment analysis. Computational linguistics.37(2),267– 307(2011).

[6] Pang, Bo; et al. Thumbs up? Sentiment Classification using Machine Learning Techniques. Empirical Methods in Natural Language Processing. ,79-86(2002).

[7] Mullen, Tony; et al. Incorporating topic information into sentiment analysis models. Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. ,25–es(2004).

[8] Kim, Yoon, Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). ,1746–1751(2014).

[9] Dong, Li; et al. Adaptive Multi-Compositionality for Recursive Neural Network Models. IEEE/ACM Transactions on Audio, Speech, and Language Processing.24(3) , (2015).

[10] Yang, Zichao; et al. Hierarchical attention networks for document classification. Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. ,1480–1489(2016).

[11] Vaswani, et al. Attention is All You Need. Proceedings of the 31st International Conference on Neural Information Processing Systems. ,6000–6010(2017).

[12] Zhang, Shunxiang; Wei, Zhongliang; Wang, Yin; Liao, Tao, Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary. Future Generation Computer Systems.81 ,395–403(2017).

[13] Lilleberg, Joseph; Zhu, Yun; Zhang, Yanqing, Support vector machines and word2vec for text classification with semantic features. 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC). ,136–140(2015).

[14] Dai, Yuanfei; et al. Relation Classification via LSTMs Based on Sequence and Tree Structure. IEEE Access. 6 ,64927–64937(2018).

[15] Yang, Xiaoyilei; et al. Sentiment Analysis of Weibo Comment Texts Based on Extended Vocabulary and Convolutional Neural Network. Procedia Computer Science. 147 ,361–368(2019).

[16] Chuhan, Wu; et al. NPA: Neural News Recommendation with Personalized Attention. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD. ,2576– 2584(2019).

[17] Laurens van der, Maaten; et al. Visualizing data using t-sne. Journal of machine learning researcn. 9(86),2579–2605(2008).