Text Sentiment Analysis Based on a Dynamic Pruning Capsule Network

Hankiz Yilahun, Peiliang Zhang(✉), Mijit Ablimit, and Askar Hamdulla

Xinjiang University, Xinjiang, China
1334195383@qq.com

Abstract. Dynamic pruning of capsules and a variable weight BIGRU model (DP-CAPS-VW-BIGRU) are proposed in this paper to address the problem of unnecessary capsule connections that cause noise and the insufficient utilization of bidirectional GRU network information. The input text is vectorized using the ERNIE pretraining model, and the convolutional layers are used to extract context information from the text. The main capsule layer uses vector output instead of the scalar output of the convolutional network to preserve instantiation parameters, and the convolutional capsule layer further extracts deeper feature information. To reduce the impact of noise caused by unnecessary capsule connections on model performance, dynamic pruning of the capsule network is proposed in this paper. At the same time, a variable weight bidirectional GRU network is proposed to improve the utilization of forward and backwards information. Experimental results show that the proposed model can effectively improve the performance of text sentiment analysis tasks.

Keywords: Capsule Network · Dynamic Pruning · Sentiment Analysis

1 Introduction

Currently, people can freely express all kinds of content and sentiment on social media every day. And Weibo has become the main internet platform for most Chinese people in their everyday life, for communicating, and obtaining various information [1]. Therefore, research on Weibo comments has become increasingly important. Based on previous research, this paper proposes a sentiment analysis model based on a dynamic pruning capsule network and variable weight bidirectional GRU. This method can reduce the problem of noise caused by unnecessary connections between capsules, and simultaneously improve the context utilization, thus effectively improveing the model performance and providing new solutions for sentiment analysis researchers.

2 Research Status

There are currently two mainstream methods for sentiment analysis: sentiment dictionaries and machine learning/deep learning. In research on emotion dictionary methods, Whissell C [2] proposed a classification method based on sentiment dictionary for the

M. Zhang et al. (Eds.): CCF NCCA 2023, CCIS 1959, pp. 148–160, 2024.
https://doi.org/10.1007/978-981-99-8764-1_12

first time. He used digit-marked sentiment words to generate sentiment dictionaries and then matched them with the corpus text to achieve sentiment classification. Li Tong et al. [3] used the concept of "center words" to solve complex information extraction tasks and evaluated important information in the text using statistical analysis. Thelwall M et al. [4] added words and phrases that appeared multiple times in social commentary texts to the sentiment dictionary and achieved sentiment classification for informal commentary texts posted on social media. Pan Minghui [5] specifically constructed two Weibo sentiment word dictionaries for her subsequent research tasks, formulated corresponding sentiment classification rules for the two dictionaries, and finally used SVM to achieve a six-class sentiment task. Saif H et al. [6] took the co-occurrence patterns of vocabulary in different contexts as factors affecting sentiment polarity determination, and updated the polarity or intensity values of sentiment words in different contexts. In 2016, Li Yuqing [7] constructed a bilingual sentiment dictionary and improved the generalization performance of the dictionary by using relative entropy and a Gaussian mixture model.

In the field of traditional machine learning research, Bo Pang et al. [8] used three machine learning methods (Naive Bayes, Maximum Entropy Classification, and Support Vector Machine) for emotion task classification, eliminating the need to be limited by emotion dictionaries and achieving more free and flexible emotion classification. Sharma A et al. [9] successfully improved the performance of SVM by integrating "weak" support vector machine classifiers using boosting algorithm classification performance. Jiang et al. [10] selected five specific topics on Twitter and used support vector machines to sentiment classify Twitter texts on a particular topic. Dragoni M et al. [11] used the overlap of possible conceptual domains to construct a universal model that can calculate the polarity of text belonging to any domain. Sentiment words were extracted from the text and then SVM models were used for sentiment classification, dividing sentiment into different categories based on weights.

In the field of deep learning, many researchers have applied deep learning technology to text sentiment analysis. Ghorbani M et al. [12] proposed an integrated structure that combines CNN and LSTM networks to accurately determine the polarity of emotions and viewpoints. Basiri M. E et al. [13] used a neural network model with an improved attention mechanism for text sentiment analysis, showing good sentiment classification performance on the Twitter dataset. Luo Fan et al. [14] combined CNN with RNN to propose the HRNN-CNN model for text sentiment analysis. The model introduced a sentence layer in the middle layer to solve the problem of difficult feature extraction when feature information is far away, and combined convolutional neural networks to achieve cross-language information extraction. Wang et al. [22] skillfully merged monolingual and bilingual information words into vectors within their document representations. They employed the innovative bilingual attention network (BAN) model to seamlessly integrate attention vectors and achieve accurate emotion classification. Tang et al. [16] proposed a model that uses CNN, LSTM, and gated recurrent neural networks to learn sentence representation for the first time, and uses Gated RNN to achieve adaptive encoding of sentence semantics. Yang et al. [17] made a groundbreaking contribution to text classification by introducing capsule networks. They developed two architectures, Capsule-A and Capsule-B, and proposed three strategies to ensure a stable dynamic routing process. These strategies effectively mitigate interference from "background" information or noise capsules that may not have been adequately trained. Kim et al. [18] proposed a static routing method to simplify the complexity of dynamic routing calculations. Ren et al. [19] proposed a composite weighted encoding method as an alternative to traditional embedding layers. They utilized a routing algorithm based on k-means clustering theory for text classification to fully explore the relationships between capsules. DONG et al. [20] proposed a capsule network model called caps-BiLSTM for sentiment analysis. This method outperformed traditional machine learning methods and deep learning models on the MR, IMDB, and SST datasets.

3 DP-CAPS-VW-BIGRU Model

The DP-CAPS-VW-BIGRU model proposed in this paper consists of a word embedding layer, a capsule network, and a VW-BIGRU network, as shown in Fig. 1. The input layer utilizes the pretrained ERNIE model provided by Baidu to obtain the vectorized representation of the text. The capsule network layer extracts local information features of the text, while the VW-BIGRU network extracts global information features. Finally, the text classification result is obtained through softmax.

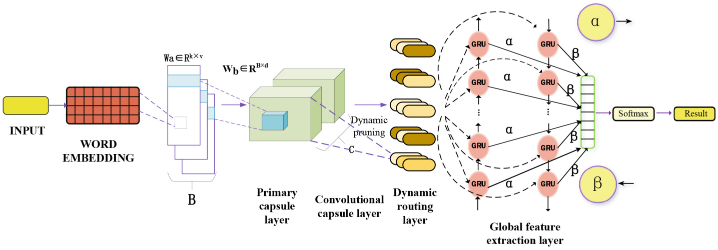

Fig. 1. DP-CAPS-VW-BIGRU Network Model Diagram

3.1 Input Layer

After performing data cleaning on the raw comment text, the cleaned text is then normalized and inputted into the pretrained ERNIE model. The text data entering the ERNIE layer are represented as $X = (X1, X2, \ldots, XN)$, where XN represents the Nth text.

3.2 Word Embedding Layer

ERNIE [21] is a knowledge-enhanced pretrained language model proposed by Baidu in 2019. During training, BERT uses a mask mechanism based on characters, ignoring the relationships between characters. The ERNIE model incorporates a wealth of prior semantic knowledge, enhancing the model's semantic representation capability. The specific mask mechanism is shown in Fig. 2. In this article, ERNIE is used as the word embedding layer.

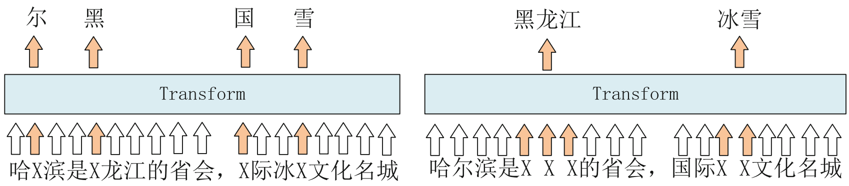

Fig. 2. Comparison of BERT and ERNIE Mask Mechanisms (Left BERT and Right ERNIE)

3.3 Capsule Network

Convolutional Layer. The convolutional layer extracts n-gram features at different positions of the ERNIE output sequence $\mathrm{E} \in \mathrm{R}^{\mathrm{L} \times \mathrm{V}}$ using convolution filters. Here, L represents the length of the text, V represents the dimension of the word embedding, and the size of the convolutional kernel $\mathrm{W_a} \in \mathrm{R}^{\mathrm{K} \times \mathrm{V}}$ represents the size of the n-gram features to be extracted. The convolutional layer has B convolutional kernels with a stride of 1. m_i^a is the i-th feature vector generated by kernel W_a at the sequence $e_{i:i+k-1}$, and the calculation formula is shown in Eq. (1).

$$m_i^a = Relu\,(e_{i:\ i+k-1} \oplus Wa + b1) \tag{1}$$

where, $\oplus$ represents the convolution operation, b_1 is the bias term, and M is the generated feature mapping matrix, $M = [m_1, m_2, \ldots, m_B] \in R^{(L-K+1) \times B}$ where $a \in \{1, 2, \ldots, B\}$, $m_a \in R^{(L-K+1)}$.

Primary Capsule Layer. In this network layer, the capsule network uses vector output instead of the scalar output of the convolutional network to preserve instantiation parameters. u_i is the initialized capsule vector, which is generated by sliding a filter of size $w_b \in R^{B \times b}$ over different vectors Mi, $i \in \{1, 2, \ldots, L - k + 1\}$. The calculation formula is shown in Eq. (2):

$$u_i = squash\,(M_i \otimes w_b + b2) \tag{2}$$

Among them, $u_i \in R^d$, d represents the capsule dimension, and the primary capsule layer has C filters that generate $u \in R^{(L-K+1) \times d}$ on each feature vector. Therefore, the number of initialized capsules generated is $(L - K + 1) \times C$. The generated capsule matrix $U = [U_1, U_2, U_3, \ldots, U_C] \in R^{(L-K+1) \times C \times d}$, where, $U_C = [u_1, u_2, u_3, \ldots, u_{(\mathrm{L-K+1})}] \in R^{(L-K+1) \times d}$.

Convolutional Capsule Layer. Similar to convolutional layers. Using Z transformation matrices to perform the capsule convolution operation on U, the capsule matrix J $= [J_1, J_2, J_3, \ldots, J_E] \in R^{(L-K-K1+2) \times Z \times d}$ is obtained, $J_E = [j_1, j_2, j_3, \ldots, j_{L-K-K1+2}] \in R^{(L-K-K1+2) \times d}$. Figure 3 visually displays the structure of each layer in the capsule network.

Dynamic Pruning Routing. In this paper, capsule network improvement occurs between the convolutional capsule layer and the dynamic routing layer. With increased dynamic routing iterations, certain capsule connections may weaken (coupling coefficients decrease), typically containing noise or irrelevant information to the model.

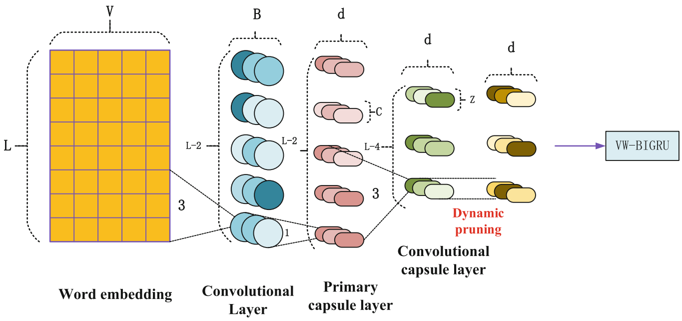

Fig. 3. DP-CAPS-VW-BIGRU Network Model Diagram

Therefore, in the last routing process of dynamic routing, this paper sets a threshold for the dynamic coupling coefficient. When the coupling coefficient is less than this threshold, the corresponding capsule is directly discarded, reducing the transmission of useless information to the parent capsule. At the same time, since different capsules represent different levels of importance, the threshold size is set as a parameter updated with the model in this paper. The dynamic pruning algorithm is shown in Table 1.

Table 1. Dynamic pruning algorithm flowchart.

DYNAMIC PRUNING ALGORITHM
Procedure ROUTING（u_j， r， t）
Initialize the coupling coefficients $b_i \leftarrow 0$
$$C_i \leftarrow softmax\ (b_i)$$
For r=1 To R do：
  for all capsule i in layer t and capsule j in t+1
  $C_{j|i} = softmax\ (b_i^r)$
  Initialization：threshold value ← 0.005
  If r=R：
   If $C_{j|i}$<threshold value：
    $C_{j|i}$=0
   Update threshold value
  For all capsule j in layer t+1：
$$s_j = \Sigma C_{j|i} u_j$$
$$v_j = squash\ (s_j)$$
Return v_j

Where $U_j \in \mathrm{R}^{\mathrm{d}}$ represents the prediction vector, $C_{j|i}$ represents the updated coupling coefficient during the dynamic routing process, where its magnitude indicates the probability of connection between capsules, s_j represents the output of the t + 1 layer parent capsule, b_i^r represents the weight after iterative updates, and the calculation formulas are as follows:

$$b_i^r = b_i^{r-1} + a_r u_j \tag{3}$$

$$a_r = squash\left(s_j\right) \tag{4}$$

VW-BIGRU Layer. Traditional bidirectional GRU networks output concatenated forward and backwards extracted features with equal weights. Due to the unevenness of text features, the feature information represented by a text from backwards to forward is different from that from forward to backwards. Therefore, in this paper, a trainable weight is added to both the forward and backwards outputs of the bidirectional GRU, allowing the model to better learn different aspects of the input sequence and better capture different aspects of the sequence. Figure 4 illustrates the structure of VW-BIGRU.

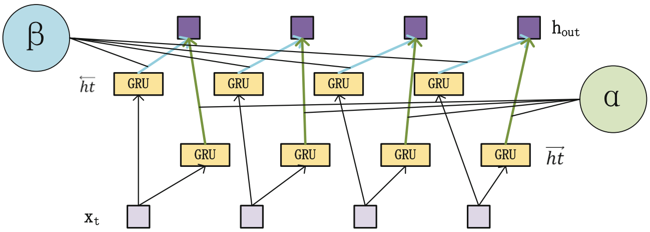

Fig. 4. VW-BIGRU Structure Diagram

In Fig. 4, α represents the weight of the forward GRU output, and β represents the weights output by the backwards GRU. In this article, these two parameters are constantly updated with the gradient, so they are called variable weight bidirectional GRU networks. The output of traditional bidirectional GRU networks is shown in Eq. (5):

$$h_{out} = \overrightarrow{h_t} + \overleftarrow{h_t} \tag{5}$$

Variable weight bidirectional GRU network with updatable weights α, β Empowering the model with stronger ability to utilize previous and subsequent information, the output of VW-BIGRU is shown in Eq. (6):

$$h_{out} = \alpha * \overrightarrow{h_t} + \beta * \overleftarrow{h_t} \tag{6}$$

Output layer: The forward and backwards hidden layers are spliced and sent to the full connected layer. Due to the large amount of data, to reduce the risk of "overfitting", the full connected layer introduces dropout to reduce the dimension of the data. The output layer classifier uses the softmax function for text classification and obtains text sentiment analysis results.

4 Experiment Settings

4.1 Data Set

This article selects two publicly available datasets for testing. Dataset 1 contains a total of 119988 Weibo comment texts, including 59993 texts with positive polarity (labelled 1) and 59995 texts with negative polarity (labelled 0). Dataset 2 contains two categories: universal and epidemic, each containing six emotions: happy (labelled 1), angry (labelled 0), sad (labelled 2), fearful (labelled 4), surprised (labelled 5), and emotionless (labelled 3). The sample size of each category is shown in Table 2.

Table 2. Number of categories in Dataset 2

	Train	Dev	Test
Usual	27768	2000	5000
Virus	8606	2000	3000

4.2 Data Preprocessing

Some special characters in the original text can impact the accuracy of the model. In this paper, text cleaning was performed on Dataset 1 and Dataset 2, and a portion of the cleaned text is shown in Table 3.

Table 3. Partial text after cleaning

Let us continue to develop delicious food during our break haha [hee hee] (dataset 1)	1
The girl sitting next to me started dating in October and received her certificate today (dataset 2)	5
Prevention and control of the epidemic. We always keep an eye on the latest developments together. Let us work together to prevent and control the epidemic. We must be vigilant, not underestimate, and take protective measures seriously. Let us work together. (dataset 2virus)	1

Due to the uneven distribution of some text on dataset 2. Considering the impact of uneven label distribution on the model, this article uses EDA [22] technology to expand some of the text in dataset 2. The number of texts before and after expansion is shown in Fig. 5.

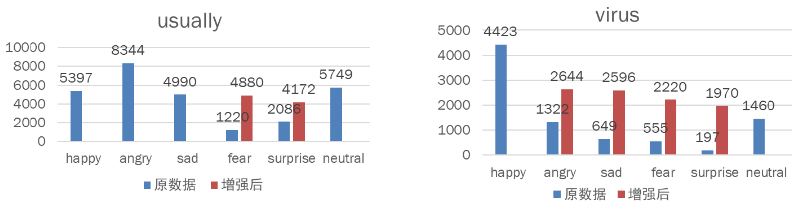

Fig. 5. Number of texts before and after data enhancement

4.3 Parameter Settings

In this experiment, the ERNIE1.0 model proposed by Baidu was used as the pretrained model. The capsule network had a filter size of 3 with 32 filters. The primary capsule layer had 32 filters with a capsule size of 16. The convolutional capsule layer had a window size of 3 with 16 filters and a size of 16. The BIGRU hidden layer had a dimension of 128, and the number of dynamic routing iterations (R) was set to 3. Additional parameters are listed in Table 4 (Table 5).

Table 4. Related model parameters

Parameter	Value
Batch size (dataset1)	25
Batch size (dataset 2)	32
Learning rate (dataset 1)	5e–5
Learning rate (dataset 2)	1e–5
Embedding Hidden size	768
dropout	0.1
Epoch	5

4.4 Evaluating Indicator

This article uses the accuracy A and F1 values on the test set as evaluation indicators, and the specific calculation formula is as follows:

$$P = \frac{T_P}{T_P + F_P} \quad R = \frac{T_P}{T_P + F_N} \tag{7}$$

$$F1 = \frac{2PR}{P + R} \quad A = \frac{T_P + T_N}{T_P + F_N + T_N + F_P} \tag{8}$$

In the above formulas, T_P represents the number of true positive cases (actual positive and predicted positive), F_P represents the number of false positive cases (actual negative but predicted positive), T_N represents the number of true negative cases (actual negative and predicted negative), and F_N represents the number of false negative cases (actual positive but predicted negative).

Table 5. The results of Various Models (%)

Model	weibo_senti_100k		smp2020			
			usual		virus	
	A	F1	A	F1	A	F1
TEXTCNN	95.41	95.39	72.12	67.33	72.59	56.48
CNN-LSTM	95.63	95.63	76.37	73.21	76.13	61.01
Capsule-A	95.54	95.54	75.83	72.35	76.29	61.32
G-Caps	96.13	96.13	76.09	72.36	76.81	61.76
Capsule-BILSTM	95.71	95.70	75.52	71.23	75.83	60.23
Capsule-VW-BIGRU	**96.44**	**96.44**	**76.86**	**73.44**	**76.53**	**61.74**
PC-Capsule-VW-BIGRU	**98.12**	**98.11**	**78.74**	**75.81**	**78.63**	**63.92**

5 Experimental Results and Discussion

5.1 Experiment 1

TEXTCNN [23]: The core idea of TextCNN is to use different-sized convolutional kernels on the input text to capture various local information.

CNN-LSTM [24]: The CNN-LSTM model initially extracts local features from the text using convolutional layers. Then, a max pooling layer is used to pool each convolutional feature into a fixed-sized vector. These vectors are then fed into an LSTM layer, which effectively handles the temporal information of the text and converts it into a fixed-sized vector representation.

Capsule A: The core of the capsule A model is a network structure composed of multiple capsule layers and fully connected layers. This model can avoid information loss caused by pooling operations in traditional convolutional neural networks.

G-Caps [25]: Yang et al. proposed a sentiment analysis model based on a unidirectional GRU and capsule feature fusion.

5.2 Experiment 2

To compare the impact of dynamic routing times on model performance in convolutional capsule layers, experiments were conducted at times 2, 3, 4, and 5. Figure 6 shows the experimental results.

5.3 Experiment 3

To verify the impact of setting thresholds on model performance in dynamic routing, experiments were conducted with threshold sizes of 0 (without improving dynamic routing), 0.01, 0.015, 0.02, 0.03, and auto (updated with the model). Figure 7 shows the experimental results.

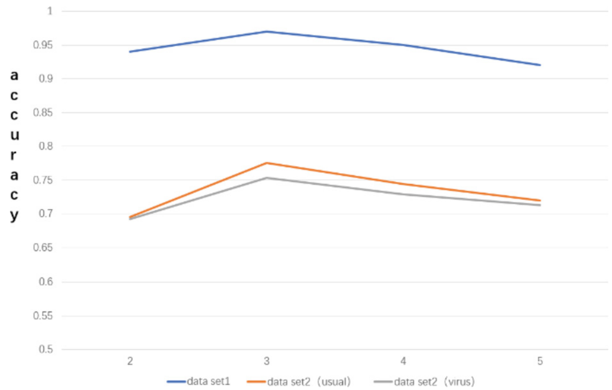

Fig. 6. Impact of Iteration Times on Model Performance

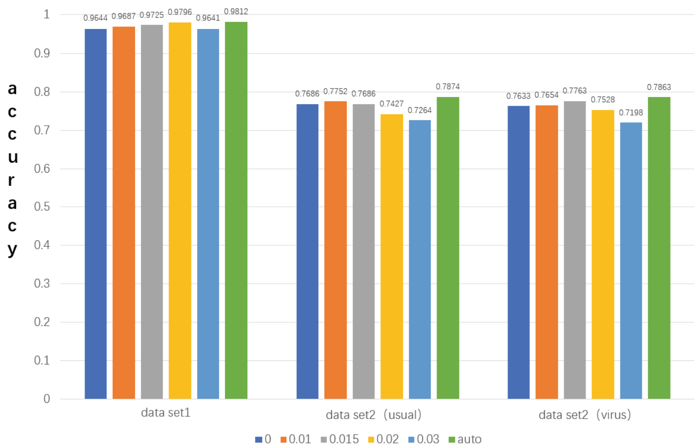

Fig. 7. Impact of Threshold on Model Performance

5.4 Result Analysis

The results of experiment 1, show that compared with TEXTCNN, Capsule-A has improved accuracy on both datasets, which demonstrates that the capsule network, using dynamic routing instead of pooling, can effectively reduce information loss. Compared with the baseline model Capsule-A, the DP-Capsule-VW-BIGRU model achieved improvements of 2.58% and 2.57% in accuracy and F1 value, respectively, on dataset 1; 2.91% and 3.46% on the "usual" category of dataset 2; and 2.34% and 2.60% on the "virus" category of dataset 2. This indicates that adding a bidirectional GRU network behind the capsule network can compensate for the capsule network's difficulty in extracting contextual information. Compared with the CNN-LSTM model, the effect of the capsule network in handling multigranularity sentiment classification tasks is significantly better than that of traditional convolutional neural networks. This also indicates that some inconspicuous feature information will be discarded by the pooling layer in traditional convolutional neural networks when processing multigranularity sentiment analysis tasks. Finally, it can be seen that the VW-BIGRU proposed in this article performs better than the traditional bidirectional GRU, which also indicates that traditional bidirectional GRU networks have insufficient utilization of forward and backwards information.

Experiment 2 was conducted to compare the impact of the number of dynamic routing iterations on model performance. The results in the figure, show that when the number of dynamic routing iterations is set to 3, the model achieves the best classification performance. As the number of routing iterations increases, the model performance decreases, which also confirms that some useless information will be generated during the dynamic routing process, and its transmission will ultimately affect the model's performance.

Experiment 3 was conducted to compare the impact of the threshold value set during dynamic routing on model classification performance. Overall, when an appropriate threshold value is set, the model's performance can be improved to some extent. Specifically, on dataset 1, when the threshold value is set to 0.02, the model's accuracy is improved by 1.52% compared with not setting a threshold (threshold value of 0); on dataset 2 (usual), when the threshold value is set to 0.01, the model's accuracy is improved by 1.52% compared with not setting a threshold; on dataset 2 (virus), when the threshold value is set to 0.015, the model's accuracy is improved by 1.30% compared with not setting a threshold. It can also be seen that as the threshold value increases, the model's classification performance tends to decrease, indicating that a too-large threshold will cause some important features of the text to be discarded, thus reducing the model's classification performance. As different capsules represent different levels of information importance, it is not reasonable to set a single threshold value for all capsules. It shows that when the threshold value is updated as a parameter during model gradient descent, the model's performance reaches its best. Therefore, in this paper, the threshold value is also updated as a parameter during model training, rather than as a hyperparameter.

6 Conclusion

In this paper, a text sentiment classification model based on capsule networks is proposed. The capsule network is used to address the problem of information loss in the pooling process of convolutional neural networks and improve the representation of detailed information. Additionally, to reduce the propagation of irrelevant information (noise) between subcapsules and parent capsules, a dynamic pruning strategy is employed during the dynamic routing process to avoid this issue. Experimental results demonstrate that the proposed model achieves good classification performance on the Weibo_senti_100k and smp2020 datasets. Future work could consider using multiple window sizes for the convolutional kernels in the capsule network layer to extract multigram syntactic information from the text.

Acknowledgements. This work was supported by the provincial and ministerial key project of the 14th Five Year Scientific Research Plan of the State Language Commission in 2022 (No. ZDI145–58) and National Social Science Fund Project (No.22XYY048).

References

1. Zhou, C., Yan, X., Yu, Z.T., Hong, X.D., Xian, Y.T.: Weibo new word recognition combining frequency characteristic and accessor variety. J. Shandong Univ. (Nat. Sci.) **50**(03), 6–10 (2015)
2. Whissell, A.C.: Objective analysis of text: II. using an emotional compass to describe the emotional tone of situation comedies. Psychol. Rep. **82**(2), 643–646 (1998)
3. Li, C., Fu, Y., Cao, Y.D.: Text sentiment classification based on phrase patterns. Comput. Sci. **04**, 132–134 (2008)
4. Thelwall, M., Buckley, K., Paltoglou, G., et al.: Sentiment strength detection in short informal text. J. Am. Soc. Inform. Sci. Technol. **61**(12), 2544–2558 (2010)
5. Pan, M.: Dictionary Based Sentiment Analysis on Chinese Weibo. Nanjing University of Aeronautics and Astronautics (2014)
6. Saif, H., He, Y., Fernandez, M., et al.: Contextual semantics for sentiment analysis of Twitter. Inf. Process. Manage. **52**(1), 5–19 (2016)
7. Li, Y., Li, X., Han, X., et al.: A bilingual dictionary based multi category emotion analysis method for weibo. J. Electr. Sci. **44**(9), 2068–2073 (2016)
8. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. arXiv (2002)
9. Sharma, A., Dey, S.: A boosted SVM based ensemble classifier for sentiment analysis of online reviews. ACM SIGAPP Appl. Comput. Reviewed **13**(04), 43–52 (2013)
10. Jiang, H.L, Yu, M., Zhou, M., et al.: Target-dependent twitter sentiment classification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 151–160 (2011)
11. Dragoni, M., Petrucci, G.: A fuzzy-based strategy for multidomain sentiment analysis. Informational J. Approximate Reasoning **93**(10), 59–73 (2018)
12. Ghorbani, M., Bahaghighat, M., Xin, Q., et al.: ConvLSTMConv network: a deep learning approach for sentiment analysis in cloud computing. J. Cloud Comput. Adv. Syst. Appl. **9**(1), 1–12 (2020)
13. Basiri, M.E., Nemati, S., Abdar, M., et al.: ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. Futur. Gener. Comput. Syst. **115**, 279–294 (2021)
14. Luo, F., Wang, H.: Chinese text sentiment classification using RNN and CNN hierarchical networks. J. Peking Univ. (Nat. Sci. Ed.) **54**(03), 459–465 (2018). https://doi.org/10.13209/j.0479-8023.2017.168
15. Wang, Z., Zhang, Y., Lee, S., et al.: A bilingual attention network for code-switched emotion prediction. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1624–1634 (2016
16. Tang, D., Qin, B., Liu, T.: Document modelling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1422–1432 (2015)
17. Yang, M., Zhao, W., Ye, J., et al.: Investigating capsule networks with dynamic routing for text classification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3110-3119 (2018)
18. Kim, J., Jang, J., Chois, S., et al.: Text classification using capsules. Neuro comput. **376**, 214 (2020)
19. Ren, H., Lu, H.: Compositional coding capsule network with k-means routing for text classification. Pattern Recogn. Lett. **160**, 1–8 (2022)
20. Dong, Y., Fu, Y., Wang, L., et al.: A sentiment analysis method of capsule network based on BiLSTM. IEEE access **8**, 37014–37020 (2020)

21. Sun, Y., Wang, S.H., Li, K., Wu, H.: ERNIE: enhanced representation through knowledge integration [EB/OL](2019−04−19)[2019−12−23]. https://arXiv.org/abs/1904.09223v1
22. Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
23. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, pp. 1746–1751 (2014)
24. Wang, J., Yu, L.C., Lai, K.R., et al.: Dimensional sentiment analysis using a regional CNN- LSTM model. In: Proceedings of the 54th Annual Meeting of the Association For Computational Linguistics (Volume 2: Short papers), pp. 225–230 (2016)
25. Yang, Y., Sun, J., Song, G.: Text sentiment analysis based on gated recurrent unit and capsule features. J. Comput. Appl. **40**(9), 2531–2535 (2020)