A Novel Chinese Sarcasm Detection Model Based on Retrospective Reader

Lei Zhang[1,2], Xiaoming Zhao[1], Xueqiang Song[1], Yuwei Fang[1], Dong Li[4], and Haizhou Wang[3(✉)]

[1] College of Computer Science, Sichuan University, Chengdu 610065, China
[2] Institute for Industrial Internet Research, Sichuan University, Chengdu 610065, China
[3] School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China
whzh.nc@scu.edu.cn
[4] Department of Computer Technology and Applications, Qinghai University, Xining 810016, Qinghai, China

Abstract. Sarcasm is a subtle form of language in which people express the opposite of what is implied. Existing research works for Chinese sarcasm detection focused on extracting features of target texts. However, there is a lot of contextual information on online social networks, which is insufficient to detect sarcasm based only on target texts. In this paper, we construct a large-scale Chinese sarcasm dataset with contextual information. Meanwhile, a sarcasm detection method based on deep learning is proposed. We used a retrospective reader in the detection process, which includes two parallel modules: Sketchy Reading and Intensive Reading. The Sketchy Reading module reads the target text and contextual information to get an initial impression. The Intensive Reading module uses a hierarchical method to get an intensive impression. Finally, we integrate the two parts to get the final prediction. Evaluation results on the dataset demonstrate the efficacy of our proposed model and the usefulness of contextual information for Chinese sarcasm detection. The research in this paper provides methods and ideas for future work in Chinese sarcasm detection on other social networking platforms.

Keywords: Sarcasm detection · Chinese · Contextual information · Retrospective reader · Online social network

1 Introduction

Merriam Webster defines sarcasm as "a mode of satirical wit depending for its effect on bitter, caustic, and often ironic language that is usually directed against an individual". It can not only disguise the hostility of the speaker but also enhance the effect of mockery or humor on the listener [2]. Because of these characteristics of sarcasm, people often use sarcasm to express their strong emotions on social media. Automatic sarcasm detection plays a significant role in various applications

© Springer Nature Switzerland AG 2022
B. Þór Jónsson et al. (Eds.): MMM 2022, LNCS 13142, pp. 267–278, 2022.
https://doi.org/10.1007/978-3-030-98355-0_23

that require the knowledge of people's sentiment or opinions [3], such as customer service, political stance detection and user intent recognition.

Existing research works for Chinese sarcasm detection mainly focused on extracting features of target texts and proposed some supervised algorithms [4,5]. However, the use of sarcasm also relies on context, which involves the presumption of commonsense and background knowledge of an event [6]. It is very difficult to determine the true intention of the speaker by only focusing on the target text. Therefore, contextual information is crucial for the task of sarcasm detection.

In this paper, inspired by the task of machine reading comprehension [1], we propose a deep learning model based on the retrospective method for sarcasm detection, that utilizes both target text and contextual information. Firstly, we take the target texts and contextual information as Input, and use the pre-trained Chinese word vector [7] for encoding. Secondly, we use two parallel modules to process the Input, namely Sketchy Reading module and Intensive Reading module. Sketchy Reading module uses the attention mechanism to simulate human behavior, and get the initial impression of the Input. Intensive Reading module uses a hierarchical method to process the Input. The Input is first passed through a recurrent layer to extract the temporal features of both target texts and contextual information, and we summarize the temporal features of contextual information. The summarized contextual information is then convoluted with the temporal features of target text to get the result. Finally, we concatenate the results of the two modules and send it to the fully connected layer to detect whether the target text is sarcastic or not. Our main contributions are summarized as follows:

Firstly, most existing sarcasm annotation corpus is in English but few in Chinese, which is a significant barrier to undertake sarcasm detection research on Chinese scenarios [25–27]. In this paper, we construct a large-scale Chinese sarcasm dataset with contextual information from 108,641 comments, which includes 2,814 manual annotated sarcastic texts and 764,231 non-sarcastic texts. Secondly, we propose a deep learning model based on the retrospective method for sarcasm detection, that utilizes both target text and contextual information. On the balanced dataset, our model achieved the highest F-score of 0.6942 and Accuracy of 0.6940. On the imbalanced dataset, our method also outperforms other baselines. Thirdly, in the component ablation test, we demonstrate the importance of the Sketchy Reading module and the Intensive Reading module, and we also show the influence of contextual information on Chinese sarcasm detection.

2 Related Work

The sarcasm detection task is a relatively new research area in natural language processing and it has become a popular research area in recent years. Sarcasm detection was initially performed using rule-based approaches. Bharti et al. [11] proposed two methods for detecting sarcastic tweets. The first is a dictionary generation algorithm based on parsing, and the second is to use exclamation points when detecting. Riloff et al. [8] presented rule-based classifiers that look for a positive verb and a negative situation phrase in a sentence. Statistical feature based approaches were used for sarcasm detection. Farias et al. [9] uses features from a variety of emotional vocabulary, and they also use features such as semantic similarity, emoticons, and counterfactuals. Reyes et al. [10] uses features such as ambiguity, unexpectedness, emotional scenario, and uses unexpectedness to measure semantic relatedness. Reyes et al. [12] uses skip-gram and character n-gram features to detect sarcasm. Machine learning algorithms were also used for sarcasm detection. The majority of work in sarcasm detection earlier relied on Support Vector Machine (SVM) [13] and Logistic Regression (LR) [14]. In recent years, people have begun to use deep learning methods for sarcasm detection. Amir et al. [15] applied convolution operation on user embedding and the utterance embedding for sarcasm detection. User embedding allowed them to learn user specific context, and auxiliary features to train the convNet. Ghosh et al. [16] uses several types of LSTM networks to model the conversation context and responses. Xu et al. [3] proposed a network to extract the differences and the semantic associations between the modalities. The importance of combining contextual information for sarcasm detection has also been realized.

For Chinese sarcasm detection, related research is still limited [17]. Liu et al. [18] constructed three unbalanced datasets based on sarcastic data from Sina Weibo, Tencent Weibo and Netease Forum, respectively. They also proposed a multi-strategy integrated learning method to solve the data imbalance problem in sarcasm detection. Tang et al. [4] constructed a traditional Chinese corpus for irony detection. In their work, some common ironic patterns were also mined. These works mainly focused on extracting features of target texts, and the use of contextual information is still lacking in Chinese sarcasm detection.

3 Dataset

In this section, we construct a large-scale Chinese sarcasm dataset with contextual information from 108,641 comments, which includes 2,814 manual annotated sarcastic texts and 764,231 non-sarcastic texts.

3.1 Data Collection

Currently, most existing sarcasm annotation corpus is in English but few in Chinese, which is a significant barrier to undertake sracasm detection research on Chinese scenario [25–27]. Previous Chinese sarcasm datasets were often constructed based on Sina Weibo [4,18,28]. However, datasets constructed based on Sina Weibo are often small in scale and only contain the target text. Moreover, the distribution of sarcastic data on Sina Weibo is relatively sparse and there is not enough contextual information. Bilibili is a video sharing website in China like YouTube but with enhanced social features [29]. Bilibili has 237 million monthly active users, which means it is one of the most popular platforms in

China. We found that there is a lot of sarcastic information in many specific topics of bilibili. Therefore, we chose to construct a data set from bilibili, and we collected a total of 108,641 target texts and related contextual information. The contextual information includes the title, introduction, reply, etc. We preprocess raw data similar to [30]. Specifically, we remove invalid strings such as web links, identifiers, and extra spaces in the text, and keep the exclamation mark. The raw data is divided into two parts: the target text and the contextual information. For the long text features in the contextual information, such as content introduction, we extract 10 words as keywords.

3.2 Manually Labeling

We have five annotators. All the annotators are postgraduate students, aged between 22 and 25, and all of them are Chinese native speakers. The annotation process follows the Irony Identification Procedure (IIP) [19]. Since the understanding of sarcasm can be subjective, we defined three fine-grained classes for sarcastic ratings: 0 (sarcastic), 1 (not sarcastic), 2 (ambiguous). To ensure the annotation quality, we synthesize the opinions of five annotators for the ambiguous cases. Then we adopt the majority if more than 80% annotators vote for it, and drop the data otherwise.

4 Approach

Figure 1 shows our deep learning model based on retrospective method for sarcasm detection. Firstly, we use the Encoder to get the initial vector representation of Input, and then send Input representation to two parallel modules, i.e., Sketchy Reading module and Intensive Reading module. In the Sketchy Reading module, we use the attention mechanism and Multilayer Perceptron (MLP) to get the initial impression. In the Intensive Reading module, we use a hierarchical method to get the intensive impression. Finally, we concatenate the outputs of the two parts to get the final representation, and send it to the prediction layer to get the final result. In this section, we will introduce the various structures of our proposed model in detail.

4.1 Word Embedding

We take contextual information and target text as Input, $I = \{i_1, \ldots, i_m\}$ represents our input sequence, m is the number of input features. We use the Encoder to obtain the initial vector representation of Input. First, the Encoder divides each input feature i_j into a sequence of words $W_j = \{w_{j1}, \ldots, w_{jn}\}$, then it uses the dense Chinese word vector proposed by [7] to convert the word sequence $W_j = \{w_{j1}, \ldots, w_{jn}\}$ into a vector representation $V_j = \{v_{j1}, \ldots, v_{jn}\}$, and finally get the Input vector representation $V = \{V_1, \ldots, V_m\}$.

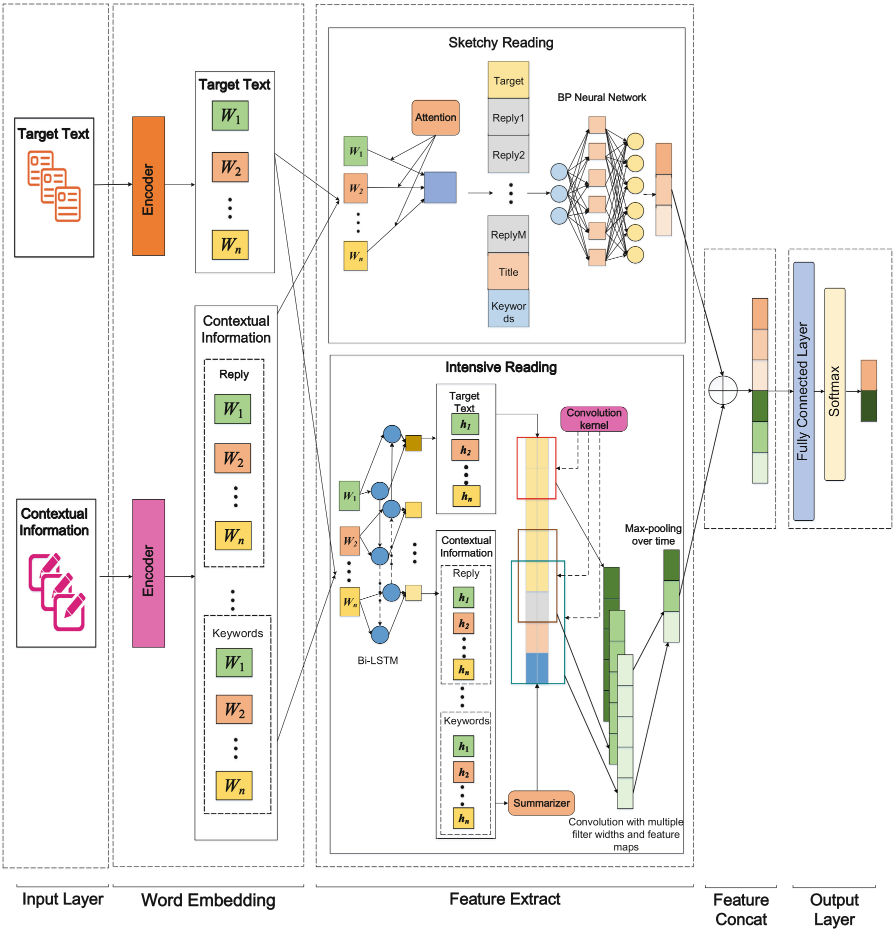

Fig. 1. Overview of our proposed model

4.2 Sketchy Reading

The purpose of the Sketchy Reading module is to get an initial impression of the Input. We use the attention mechanism to process the target text and contextual information. The vector representation V_j is passed through a two-layer neural network to obtain the attention weights α_i and construct the final feature vector representation R_j. The related equations are as follows:

$$\alpha_i = W_2 \cdot \tanh\left(W_1 \cdot v_{ji} + b_1\right) + b_2 \tag{1}$$

$$\alpha = \text{softmax}(\alpha) \tag{2}$$

$$R_j = \sum_{i=1}^{n} \alpha_i v_{ji} \tag{3}$$

Where v_{ji} is the i^{th} word embedding of the j^{th} Input feature; W_1 and W_2 are weight matrices; b_1 and b_2 are biases; n is the length of the sequence. Finally, $R = \{R_1, ..., R_m\}$ are passed through the Multilayer Perceptron(MLP), and we get the output of Sketchy Reading.

4.3 Intensive Reading

Unlike the Sketchy Reading module, in this part we pay more attention to the target text. We use a hierarchical method to process the Input, and then use CNN Layer to extract the relationship between the target text and context information, and finally get the result of Intensive Reading.

Bi-LSTM Encoder Layer. Each feature of the input is originally a sentence. Each word in the sentence is independent of other words, when the words are represented by making use of word embedding V. In this part, a new representation for each word is achieved by summarizing contextual information from both the directions in a sentence. We use a bidirectional LSTM to get a new representation of the Input. The equations of operations performed by LSTM at time step t are as follows:

$$i_t = \sigma\left(W_i \cdot x_t + U_i \cdot h_{t-1}\right) \tag{4}$$

$$f_t = \sigma\left(W_f \cdot x_t + U_f \cdot h_{t-1}\right) \tag{5}$$

$$o_t = \sigma\left(W_o \cdot x_t + U_o \cdot h_{t-1}\right) \tag{6}$$

$$\tilde{c}_t = \tanh\left(W_c \cdot x_t + U_c \cdot h_{t-1}\right) \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{8}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{9}$$

where W_i, W_f, W_o, U_i, U_f, U_o are weight matrices; x_t, h_t are input state and hidden state at time step t respectively; σ is the sigmoid function; $\odot$ denotes element-wise product.

The bidirectional LSTM is a combination of forward LSTM $\vec{h}$, which reads the sentence from w_1 to w_n, and a backward LSTM $\overleftarrow{h}$, which reads the sentence from w_n to w_1:

$$\overrightarrow{h}_t = \overrightarrow{\mathrm{LSTM}}\left(w_t, \overrightarrow{h_{t-1}}\right) \tag{10}$$

$$\overleftarrow{h}_t = \overleftarrow{\mathrm{LSTM}}\left(w_t, \overleftarrow{h_{t+1}}\right) \tag{11}$$

After this operation, for each word, we have got the forward and backward hidden state. For example, for the word t, we get $h_t = \left[\overrightarrow{h}_t, \overleftarrow{h_t}\right]$. For the target text, we expect to get as much information as possible, so we save the hidden state of all words. Finally we get the representation of the target text $H_{target} = \{h_1, h_2, \ldots, h_n\}$, where $H_{target} \in \mathbb{R}^{n \times d_{lstm}}$.

Context Summarization Layer. The size of contextual information vector after the BiLSTM encoder layer may be too large to process. For example, we now have 10 replies, the number of BiLSTM units d_{lstm} is 300 and the maximum sentence length $d_{sentence}$ is 100, then the final representation will be of size $13 \times 100 \times 300$. In order to obtain a summarized context, we send the contextual information to Context Summarization Layer. In the Context Summarization Layer, we obtain the summarized context, i.e., $H_{context} = \{h_1, \ldots, h_{m-1}\}$, where $h_i = \left\{h_i^{\text{fl}}, h_i^{\text{bl}}\right\}$, h_i^{fl} is the last forward hidden state of the i^{th} Input feature, h_i^{bl} is the last backward hidden state of the i^{th} Input feature. In this way, we obtain the word embedding that summarizes the entire sentence information, and the final output dimension is $H_{context} \in \mathbb{R}^{(m-1)\times d_{lstm}}$.

CNN Layer. In [21], the author proposed a hybrid multi-channel CNN to capture the N-grams features in a text by varying the kernel size. We take $H_{context}$ and H_{target} as input $X \in \mathbb{R}^{(m-1+n)\times d_{lstm}}$, then use the 1D convolutional layer to capture the relationship between the contextual information and the target text.

$$c_i = f\left(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b\right) \tag{12}$$

Where $\mathbf{x}_{i:i+h-1}$ refers to the concatenation of $\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+h-1}$; filter is represented by $\mathbf{w} \in \mathbb{R}^{h\times d_{lstm}}$; h is the number of features of X used to generate new features; b is a bias term and f is a non-linear function. Filter is applied to the sequence X to produce a feature map $\mathbf{C} = \{c_1, c_2, \ldots, c_{m+n-h}\}$. In this section, we use filters with window sizes of 2, 3, and 4 to capture different N-gram features. Then the feature map $\mathbf{C}$ is sent to the MAXPOOL layer to get the most relevant feature $\hat{c}$ among the N-gram features. Finally, the most relevant features are concatenated together as the output of the Intensive Reading module.

4.4 Output Layer

We concatenate the outputs of the two parallel modules and send it to the Fully Connected Layer to get the final prediction.

$$\hat{y} = \operatorname{softmax}\left(W_f \cdot O_{final} + b_f\right) \tag{13}$$

Where W_f is the weight matrix; b_f is the bias.

5 Experiments

In this section, we evaluate the performance of the proposed model for sarcasm detection. All experiments were undertook on a workstation with an Intel Core i9-10900 CPU and NVIDIA GeForce RTX 3070 GPU with 8 GB of RAM. We use Accuracy, Recall, Precision, and F-score to evaluate the performance of our detection method.

5.1 Performance Comparison with the Baseline Approaches

In this section, we constructed a balanced dataset, which includes 2,814 sarcastic data and 2,814 non-sarcastic data randomly selected. We divide the data into training set, development set and test set with a ratio of 80%:10%:10%. Meanwhile, we implemented the following baselines:

1. **Bi-LSTM.** Bi-LSTM is one of the most popular methods for solving text classification problems. It was used as a baseline model in sarcasm detection [3].
2. **Text CNN.** Text CNN is another great approach that has appreciable effects in detecting Chinese sarcastic texts [5].
3. **DPCNN.** It is difficult for Text CNN to obtain the long-distance relationship of the text through convolution, whereas DPCNN [22] can express the long-distance relationship in the text by increasing the network depth.
4. **ERNIE.** ERNIE [23] is a pre-training model proposed by Baidu. It implicitly learned the information about knowledge and longer semantic dependency. It has achieved good performance in many Chinese NLP fields.
5. **Hierarchical Attention Network.** HAN [24] uses the attention mechanism to classify documents. We form a document with contextual information and target text, and then use HAN for classification. HAN is an excellent baseline in sarcasm detection [31], which can utilize the contextual information.

Table 1 shows the performance of different deep learning models and R-net on the dataset. The Receiver Operating Characteristic(ROC) curve is shown in Fig. 2.

Table 1. The performance of different deep learning models and R-Net on the dataset.

Model	F-score	Precision	Recall	Accuracy
BiLSTM [20]	0.6141	0.6184	0.6175	0.6143
Text CNN [21]	0.5865	0.5966	0.5910	0.5938
DPCNN [22]	0.5840	0.5937	0.5907	0.5859
HAN [24]	0.6678	0.6797	0.6697	0.6743
ERNIE [23]	0.6411	0.6443	0.6415	0.6442
R-Net(our model)	**0.6942**	**0.6949**	**0.6940**	**0.6953**

From the Table 1, we can see that BiLSTM performs better than Text CNN and DPCNN, which indicates that BiLSTM have an advantage of extracting features, and that is why we use it in the Intensive Reading module. However, the methods that only focus on the target text (BiLSTM, Text CNN, DPCNN) do not perform well, and the HAN that uses the contextual information performs the best besides our method. The performance of our model is also better than the pre-trained model ERNIE. The ERNIE is likely to be affected by the pre-training data field, and cannot be perfectly applied to the field of Chinese sarcasm detection. In conclusion, our model achieved the best F-score and accuracy. In addition, the AUC score shown in Fig. 2 also proves that the performance of the method using contextual information is better than other methods that only focus on the target text, which shows that contextual information is very important for Chinese sarcasm detection.

5.2 Robustness Analysis with the Baseline Approaches

In the previous subsection, we used a balanced dataset to undertake model comparison experiment. However, in realistic scenarios, there may be only a small

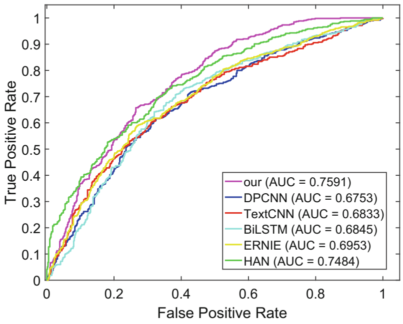

Fig. 2. The ROC curve.

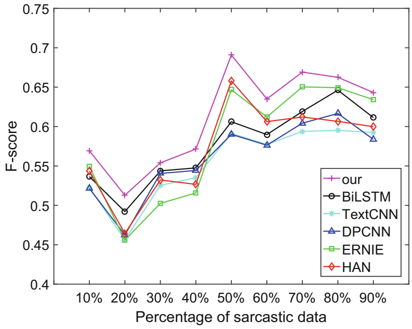

Fig. 3. Comparison of the models trained on the datasets with different percentages of sarcastic texts. The total number of texts is 3,000.

proportion of sarcastic texts. To verify the robustness of the proposed model, we tested their performances on different proportions of sarcastic texts. We fixed the capacity of our dataset to 3,000 and varied the percentage of sarcastic texts from 10% to 90% at increments of 10%. As shown in Fig. 3, it can be seen that our method and HAN achieved the best performance when the sarcastic texts accounted for 50%. However, other models get the best performance when the percentage of sarcastic texts is relatively high. When the proportion of sarcastic texts exceeds 60%, our method is closer to the performance of ERNIE. Compared with other models, our model always has a better performance. Therefore, our method is more robust than other methods in realistic scenarios.

5.3 Component Analysis of Our Model

We further evaluate the influence of Sketchy Reading module, Intensive Reading module, as well as contextual information on the final performance. The subsets of the component set can be represented by the set difference function given as

$$F \backslash F' = \{x \mid x \in F \wedge x \notin F'\} \tag{14}$$

where F is the R-Net, F' is the component of R-Net. The components we used in component ablation test are as follows:

- F: All components of the proposed model will be used.
- $F \backslash SR$: Remove Sketchy Reading module from R-Net.
- $F \backslash IR$: Remove the Intensive Reading module from R-Net.
- $F \backslash CI$: Remove contextual information from R-Net.

The evaluation results are shown in Fig. 4. We can see that both the Sketchy Reading module and the Intensive Reading module improve the performance of the model. Moreover, the Intensive Reading module are more effective than the

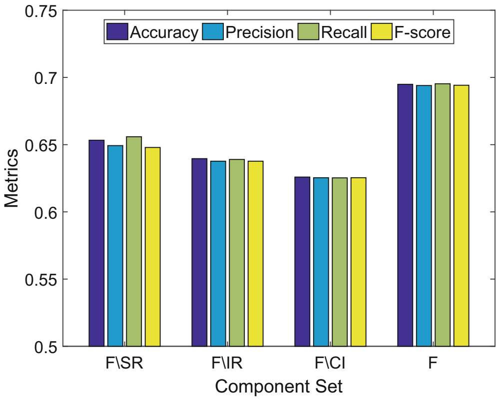

Fig. 4. The performance of different components in component ablation test.

Sketchy Reading module in the detection. This is because the intensive impression of sarcastic data is more important. In addition, when we remove the contextual information, the model performs the worst, which shows that contextual information is also very important in Chinese sarcasm detection.

6 Conclusion and Future Work

In this paper, we have contracted a Chinese sarcasm dataset with contextual information. Meanwhile, we propose a deep learning model based on retrospective method for sarcasm detection. The evaluation results demonstrate the effectiveness of our model and the usefulness of contextual information for Chinese sarcasm detection. In future work, we will incorporate auxiliary information (such as user information) in the dataset into the task of Chinese sarcasm detection, and we will also study how to use common sense knowledge in our method.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (NSFC) under grant nos. 61802270, 61802271, 81602935, and 81773548. In addition, this work is also partially supported by Joint Research Fund of China Ministry of Education and China Mobile Company (No. CM20200409), the Key Research and Development Program of Science and Technology Department of Sichuan Province (No. 2020YFS0575), and the Sichuan University and Yibin Municipal People's Government University and City Strategic Cooperation Special Fund Project (Grant No. 2020CDYB-29).

References

1. Zhang, Z., Yang, J., Zhao, H.: Retrospective reader for machine reading comprehension. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence, pp. 14506–14514 (2021)

2. Tay, Y., Luu, A.T., Hui, S.C., Su, J.: Reasoning with sarcasm by reading in-between. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 1010–1020 (2018)
3. Xu, N., Zeng, Z., Mao, W.: Reasoning with multimodal sarcastic tweets via modeling cross-modality contrast and semantic association. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3777–3786 (2020)
4. Tang, Y.J., Chen, H.H.: Chinese irony corpus construction and ironic structure analysis. In: Proceedings of the 25th International Conference on Computational Linguistics, pp. 1269–1278 (2014)
5. Lin, S.K., Hsieh, S.K.: Sarcasm detection in Chinese using a crowdsourced corpus. In: Proceedings of the 28th Conference on Computational Linguistics and Speech Processing, pp. 299–310 (2016)
6. Hazarika, D., Poria, S., Gorantla, S., Cambria, E., Zimmermann, R., Mihalcea, R.: CASCADE: contextual sarcasm detection in online discussion forums. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 1837–1848 (2018)
7. Li, S., Zhao, Z., Hu, R., Li, W., Liu, T., Du, X.: Analogical reasoning on Chinese morphological and semantic relations. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 138–143 (2018)
8. Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., Huang, R.: Sarcasm as contrast between a positive sentiment and negative situation. In: Proceedings of the 10th Conference on Empirical Methods in Natural Language Processing, pp. 704–714 (2013)
9. Farías, D.I.H., Patti, V., Rosso, P.: Irony detection in Twitter: the role of affective content. ACM Trans. Internet Technol. **16**(3), 1–24 (2016)
10. Reyes, A., Rosso, P., Buscaldi, D.: From humor recognition to irony detection: the figurative language of social media. Data Knowl. Eng. **74**, 1–12 (2012)
11. Bharti, S.K., Babu, K.S., Jena, S.K.: Parsing-based sarcasm sentiment recognition in Twitter data. In: Proceedings of the 7th IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1373–1380 (2015)
12. Reyes, A., Rosso, P., Veale, T.: A multidimensional approach for detecting irony in twitter. In: Proceedings of the 8th International Conference on Language Resources and Evaluation, pp. 239–268 (2013)
13. Ghosh, A., et al.: SemEval-2015 task 11: sentiment analysis of figurative language in Twitter. In: Proceedings of the 9th International Workshop on Semantic Evaluation, pp. 470–478 (2015)
14. Bamman, D., Smith, N.A.: Contextualized sarcasm detection on Twitter. In: Proceedings of 9th International AAAI Conference on Web and Social Media, pp. 261–265 (2015)
15. Amir, S., Wallace, B.C., Lyu, H., Carvalho, P., Silva, M.J.: Modelling context with user embeddings for sarcasm detection in social media. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, pp. 167–177 (2016)
16. Ghosh, D., Fabbri, A.R., Muresan, S.: Sarcasm analysis using conversation context. In: Proceedings of the 26th International Conference on Computational Linguistics Computational Linguistics, pp. 755–792 (2018)
17. Van Hee, C., Lefever, E., Hoste, V.: SemEval-2018 task 3: Irony detection in English Tweets. In: Proceedings of the 12th International Workshop on Semantic Evaluation, pp. 39–50 (2018)

18. Liu, P., Chen, W., Ou, G., Wang, T., Yang, D., Lei, K.: Sarcasm detection in social media based on imbalanced classification. In: Proceedings of the 15th International Conference on Web-Age Information Management, pp. 459–471 (2014)
19. Li, A.R., Chersoni, E., Xiang, R., Huang, C.R., Lu, Q.: On the "easy" task of evaluating Chinese irony detection. In: Proceedings of the 33th Pacific Asia Conference on Language, Information and Computation, pp. 452–460 (2019)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
21. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 11th Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751 (2014)
22. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 562–570 (2017)
23. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: enhanced language representation with informative entities. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1441–1451 (2019)
24. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 14th Conference of The North American Chapter of The Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489 (2016)
25. Joshi, A., Sharma, V., Bhattacharyya, P.: Harnessing context incongruity for sarcasm detection. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 757–762 (2015)
26. Oraby, S., Harrison, V., Reed, L., Hernandez, E., Riloff, E., Walker, M.: Creating and characterizing a diverse corpus of sarcasm in dialogue. In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 31–41 (2016)
27. Khodak, M., Saunshi, N., Vodrahalli, K.: A large self-annotated corpus for sarcasm. In: Proceedings of the 10th Language Resources and Evaluation Conference, pp. 641–646 (2018)
28. Xiang, R., et al.: Ciron: a new benchmark dataset for Chinese irony detection. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp. 5714–5720 (2020)
29. Yan, L., Cha, N., Cho, H., Hwang, J.: Video diffusion in user-generated content website: an empirical analysis of Bilibili. In: Proceedings of the 21st International Conference on Advanced Communication Technology, pp. 81–84 (2019)
30. Gong, X., Zhao, Q., Zhang, J., Mao, R., Xu, R.: The design and construction of a Chinese sarcasm dataset. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp. 5034–5039 (2020)
31. Srivastava, H., Varshney, V., Kumari, S., Srivastava, S.: A novel hierarchical BERT architecture for sarcasm detection. In: Proceedings of the 2nd Workshop on Figurative Language Processing, pp. 93–97 (2020)