YNU-HPCC at SemEval-2022 Task 6: Transformer-based Model for Intended Sarcasm Detection in English and Arabic

Guangmin Zheng, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
Contact: gmzheng@mail.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

In this paper, we (a YNU-HPCC team) describe the system we built in the SemEval-2022 competition. As participants in Task 6 (titled "iSarcasmEval: Intended Sarcasm Detection In English and Arabic"), we implement the sentiment system for all three subtasks in English and Arabic. All subtasks involve the detection of sarcasm (binary and multilabel classification) and the determination of the sarcastic text location (sentence pair classification). Our system primarily applies the sequence classification model of a bidirectional encoder representation from a transformer (BERT). The BERT is used to extract sentence information from both directions for downstream classification tasks. A single basic model is used for single-sentence and sentence-pair binary classification tasks. For the multilabel task, the Label-Powerset method and binary cross-entropy loss function with weights are used. Our system exhibits competitive performance, obtaining 12/43 (21/32), 11/22, and 3/16 (8/13) rankings in the three official rankings for English (Arabic).

1 Introduction

Satirical text is a rhetorical device for implicitly expressing emotions by using words that are contrary to the actual intention to achieve a satirical or humorous linguistic effect. The true semantics of satirical texts cannot be directly inferred from the text vocabulary, and contradictions exist between their literal meaning and the true intention. Therefore, the detection of sarcasm and its sentiment discrimination are more challenging in natural language processing (NLP) problems.

Task 6 in the SemEval-2022 competition is a sarcasm detection task that consists of three subtasks (Abu Farha et al., 2022).

- Subtask A: Detect sarcastic meaning from a given tweet.
- Subtask B: Identify the tweet as "no sarcasm" or one or more of the six given sarcastic speech categories.
- Subtask C: Determine the position of the satirical text from the given tweets and their non-satirical restatements (0 means the first sentence is satirical, and 1 means the second sentence is satirical).

In previous sarcasm detection tasks, researchers used supervised learning methods based on the support vector machines and logistic regression (González-Ibáñez et al., 2011) to study ironic and non-ironic tweets that directly express positive and negative views. However, these traditional machine learning methods cannot mine the deep semantic information hidden in the text. Relying only on the surface semantic information can easily result in an incorrect judgment regarding irony. In contrast, deep learning methods show excellent results in deep semantic mining. By using the context information of the text to be detected (Bamman and Smith, 2015), we can further mine the behavior information of social users. We can achieve better performance by using a bidirectional recurrent neural network to capture the syntactic and semantic information of the target tweet text and the automatic learning features of historical tweets related to the target tweet for sarcasm detection (Zhang et al., 2016). The convolutional long-term and short-term memory network (CNN-LSTM-DNN) (Ghosh and Veale, 2016) achieved remarkable results.

In this paper, we propose a deep learning system for Task 6 in SemEval-2022, titled "iSarcasmEval: Intended Sarcasm Detection In English and Arabic." We use a pre-trained bidirectional encoder representation from a transformer (BERT) (Devlin et al., 2019) sequence classification model as the base model. For single-sentence and sentence-pair binary classification tasks, we use fine-tuning methods on a basic model. We employ the Label-

Powerset (Nazmi et al., 2020) approach and train the basic model by applying a binary cross-entropy loss function with weights for the multi-label task. The contributions of this study are as follows.

- For the sentiment analysis problem, we propose a basic model using a pre-trained BERT sequence classification model.
- The use of a binary cross-entropy loss function with weights is more advantageous for performing multi-label classification with uneven label distribution task.
- Fine-tuning on multilingual datasets (Pires et al., 2019) leads to a significant improvement in the scores of prediction results.

The remainder of this paper is organized as follows. In Section 2, we describe the proposed system and model in detail. The experiments and results are discussed in Section 3. Finally, conclusions are presented in Section 4.

2 System Description

The general structure of our system consists of four modules, described as follows.

Input layer. In this layer, we build text-processing tools to perform pre-processing of text and embedding of words. A large amount of useless information in the given text, such as user (@user), URL (http://ie.com), and escape symbols (\s, \j, etc.), increases the computational effort and complexity of a model. We remove the useless information in advance by using regular expressions and convert all words to lowercase, without breaking the structure of hashtags (#hashtag). We believe that the search and tagging of these tweets depends mainly on their hashtags (Peng et al., 2018). The preservation of the hashtag structure improves the accuracy of sarcasm detection. Tokenizers provided by HuggingFace[1] are used to process information such as punctuation, emoticons, non-English (or non-Arabic) letters, numbers, and hashtags (#hashtag) that are still present in the text and to rapidly perform word separation. Tokenizers can also perform word embedding using continuous low-dimensional vectors to represent word features (Mikolov et al., 2013). In this layer, we obtain the sequence of representations to be used as inputs to the subsequent modules.

[1]https://huggingface.co/.

Context encoder. Devlin et al. proposed a new natural language representation model named BERT in 2018, which successfully achieved state-of-the-art results in 11 NLP tasks, winning a plethora of accolades from the NLP community. The model is based on a bidirectional transformer for large-scale pre-training and can be fine-tuned by users to handle different text-processing tasks. One variant of the BERT is RoBERTa (Liu et al., 2019), which enhances the effect of the BERT by improving its pre-training method without affecting the structure of the BERT. Unlike the BERT static mask approach, RoBERTa uses dynamic masks, where the tokens masked for each sequence are changed in different epochs of training. RoBERTa removes the NSP task and uses the FULL-SENTENCES training approach. RoBERTa also uses a larger mini-batch, more data, and larger number of sentences. Therefore, the RoBERTa-pretrained model yields better results. In this module, we mainly use the pre-trained BERT model and its variant RoBERTa model to complete the contextual encoder.

Fully connected layer. Fully connected networks are used for downstream classification tasks.

Output layer. The output of the fully connected layer is processed to complete the label prediction. Different tasks require different processing methods.

The details of each subtask in each module are discussed in the following.

2.1 Subtask A: Sentence Classification

The first part of this task was to extract semantic information from a given tweet text. We termed this sentence classification (Dao et al., 2020), where we predict whether a sentence is sarcastic. For this purpose, our approach generated 768-dimensional word embeddings for each word in a sentence by using a pre-trained BERT model. We selected the first token (i.e., "[CLS]") of the sentence into the sequence classification because it integrated the semantic information of the entire sentence. The word embeddings obtained from the previous step were then connected to a fully connected layer, which transformed the 768-dimensional input into two-dimensional values. These values were then fed into Softmax to calculate the probability that the sentence is sarcastic. Finally, the probability results were fed into argmax to form labels; in our experimental setup, 1 indicated sarcasm and 0 indicated non-sarcasm. The model architecture is

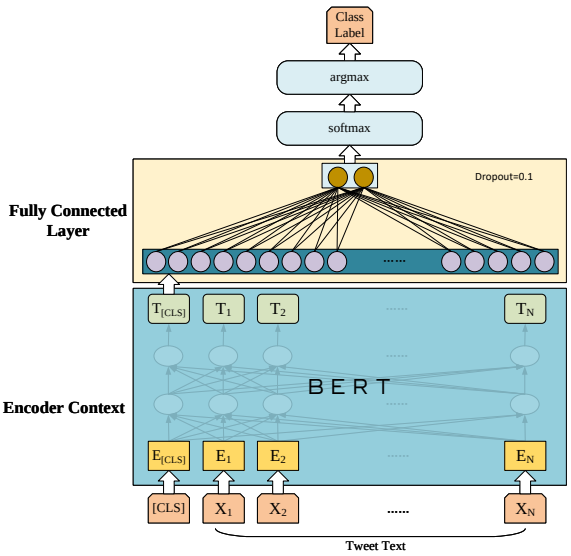

Figure 1: System of binary classification for sentence

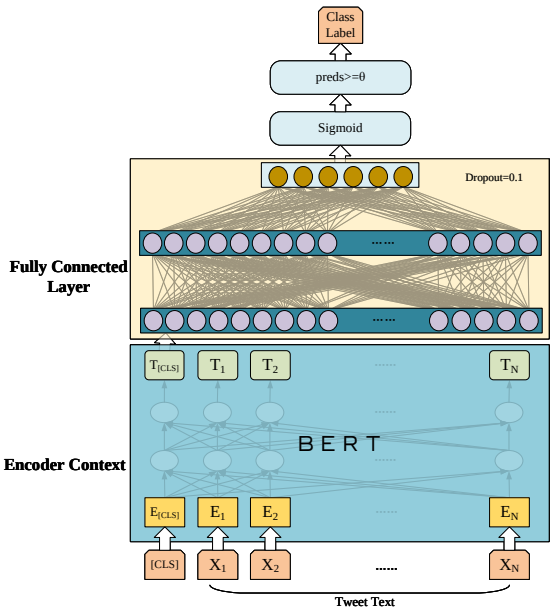

Figure 2: Multi-label classification system for sentence

Table 1: Quantity of labels by category in English dataset.

Category	Quantity
sarcasm	713
irony	155
satire	25
understatement	10
overstatement	40
rhetorical_question	101

illustrated in Figure 1.

2.2 Subtask B: Multi-Label Classification

This task was also a sentence classification task, with the difference being that it involved the prediction of multiple binary targets simultaneously based on a given input. For the upstream task of sentence information extraction, we used the pre-trained BERT model. For the downstream classification task, we used the Label-Powerset method (Tsoumakas et al., 2011), which sets the number of labels to the number of output neurons in the network. We can directly apply an arbitrary binary classification loss function to the neural network model, and the model can simultaneously output all targets. At this point, we only need to train one model; the training time is shorter, and the network can also learn the relevance of different labels through the output neurons. We counted the number of various types of labels in the English dataset (Table 1) and discovered that the number of labels is dozens of times different. To solve the problem of label imbalance, we appropriately added a network layer to improve the classification effect. We employed the RELU activation function for fast computation between two linear layers to boost the sparsity of the simultaneous network and reduce the interdependence of parameters to alleviate the overfitting problem. In addition, we utilized a binary cross-loss function with weights to focus the model on the sparse labels, as stated in Section 2.4.

We selected the values generated by the aforementioned work into sigmoid and mapped their range to (0, 1) to reply to the confidence level of each label. We set a threshold of $\theta = 0.5$, and when the predicted value was greater than or equal to θ, we considered that the prediction contained this label; otherwise, it did not. The model architecture is shown in Figure 2.

2.3 Subtask C: Sentence-Pair Classification

This task is a sentence pair classification task, similar to subtask A, but with two sentences as the input. In the input layer, in addition to the work done in Subtask A, we must split the two sentence representations using the "[SEP]" token. The inputs are then passed to the pre-trained BERT model to obtain the "[CLS]" token, which integrates the semantic information of the entire sentence. The subsequent model structure was the same as that for Subtask A. The "[CLS]" token is passed to the subsequent module to obtain the location label of

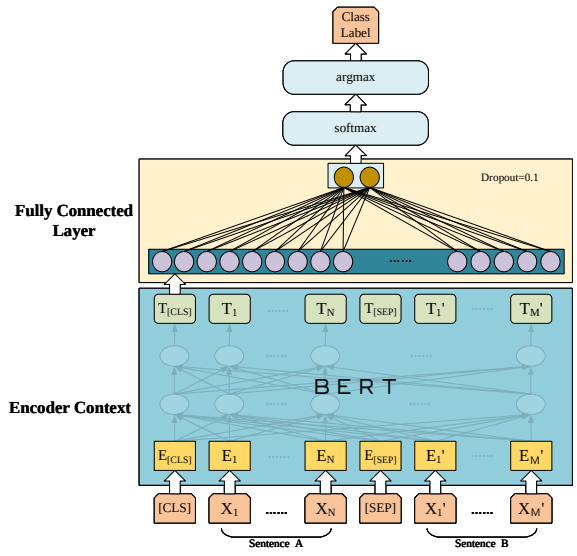

Figure 3: System of binary classification for sentence pair

the sarcastic text, with 0 indicating sentence A and 1 indicating sentence B. The model architecture is illustrated in Figure 3.

2.4 Training and Hyper-parameters

We used a binary cross-entropy loss function for subtasks A and C and a binary cross-entropy loss function with weights for subtask B to train the model. For the sample imbalance between labels in the multilabel classification task of subtask B, we introduced parameter w_i. We increased the attention of the system to scarce labels by assigning higher weights to labels that appear less frequently. For batch data D(x, y) containing N samples with M labels per sample, the loss function is calculated as follows:

$$loss = \frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} l_n, \tag{1}$$

$$l_n = \frac{1}{\mathrm{M}} \sum_{i=1}^{\mathrm{M}} l_n^i, \tag{2}$$

where l_n^i is the loss corresponding to the i-th label of the n-th sample.

$$l_n{}^i = -w_i[y_n^i \cdot \log \sigma(x_n^i) + (1 - y_n^i) \cdot \log(1 - \sigma(x_n^i))] \tag{3}$$

where σ is the sigmoid function. w_i is the weight parameter of the i-th label, which is calculated from the i-th label number, $Label_i$, by applying the following equation.

$$w_i = \frac{\sum Label}{Label_i} \tag{4}$$

For all the subtasks, we used the AdamW (Loshchilov and Hutter, 2019) optimizer to train the model with a batch size of 16.

Hyperparameters. The dimensionality (d) of the word embedding was 100, learning rate was 2e-5, weight decay was 0.01, and dropout ratio was 0.1 for all layers in all models.

3 Experimental

Datasets. The datasets used were the English dataset (3467 data) and Arabic dataset (2602 data) provided by the competition, with no other external corpus. Only one dataset in English was used for subtask B, except for subtasks A and C, in which two sub-datasets were used. We thank the organizers for their contributions to the data.

Evaluation Metrics. The main evaluation metrics are as follows:

- SubTask A: F1-score for the sarcastic class.
- SubTask B: Macro-F1 score.
- SubTask C: Accuracy.

Implementation Details. To solve the data imbalance problem, we sampled the data selected for training such that the number of 0/1 tags was as similar as possible. For the sentence pair classification task, we take the tweet text in the dataset as sentence A and the rephrase text as sentence B, and assign label 0. After switching the positions of tweet text and rephrase text, we assign label 1. We divided the training data into a training set and a development set at a ratio of 8:2. We trained our models on the training set, evaluated the predictions on the development set using evaluation measures, and saved the models with the highest evaluation scores during the process. We used the Pytorch framework provided by the Huggingface library for the pre-trained BERT model and bert-base-multilingual-uncased and roberta-base for the binary classification, multi-label classification, and sentence pair classification included in this task.

Results and Analysis. Tables 2, 3, and 4 present comparative results for each task. The bolded scores are those assessed by participating in the competition leaderboard. On the competition leaderboard, our system ranked 12/43 (21/32) in

Table 2: Comparable results of experiments for subtask A.

Model	Fine-tune	Test	F1
bert-base-multilingual-uncased	English	English	0.306
	English+Arabic		0.285
	Arabic	Arabic	0.202
	English+Arabic		0.245
roberta-base	English	English	**0.392**
	Arabic	Arabic	**0.323**

Table 3: Comparable results of experiments for subtask B.

Model	Loss	Macro-F1
6-binary	BCE	0.0702
bert-base-multilingual-uncased	BCE	0.0672
	BCEwithWeight	0.0795
	Dice	0.0379
	Focal	**0.0646**

Table 4: Comparable results of experiments for subtask C.

Model	Fine-tune	Test	Acc
bert-base-multilingual-uncased	English	English	0.815
	English+Arabic		**0.805**
	Arabic	Arabic	0.5
	English+Arabic		**0.755**
roberta-base	English	English	0.860
	Arabic	Arabic	0.5

English (Arabic) in Task A, 11/22 in Task B, and 3/16 (8/13) in Task C.

As shown in the tables, our approach achieves significant results. This is mainly because the BERT model is a multilayer bidirectional transformer encoder that can be integrated into various NLP downstream tasks and obtains the best results. In addition, the training results using the roberta-base model are significantly better, indicating that the pre-training approach of the BERT can be improved. Moreover, fine-tuning with multiple languages significantly improved the training results, particularly for Arabic. As revealed in Table 4, fine-tuning the model on the Arabic dataset individually shows worse results, which may be due to the fact that very little information is extracted from the Arabic dataset only. In the future, we can improve the method of fine-tuning in multiple-language datasets by balancing the vectors of different languages to improve the model effect.

For multi-label classification, the Label-Powerset method is not as effective as stand-alone dichotomous classifier training; however, combined with a dichotomous cross-entropy loss function with weights, the results can be slightly improved while saving a lot of training time.

4 Conclusion

In this paper, we describe a deep learning model for the sentiment analysis task in the SemEval-2022 competition (Task 6: iSarcasmEval: Intended Sarcasm Detection In English and Arabic). A BERT sequence classification model was used as the base model. The final submitted system performed admirably and ranked third in one of the rankings. However, there is still considerable potential for improvement. Therefore, in future investigations, we will attempt to extend this model with improved capabilities to achieve better outcomes.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61966038. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Ibrahim Abu Farha, Silviu Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

David Bamman and Noah A. Smith. 2015. Contextualized sarcasm detection on twitter. In *Proceedings of the 9th International Conference on Web and Social Media (ICWSM 2015)*, pages 574–577.

Jiaxu Dao, Jin Wang, and Xuejie Zhang. 2020. YNU-HPCC at SemEval-2020 task 11: LSTM network for detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1509–1515, Barcelona (online). International Committee for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to*

Subjectivity, Sentiment and Social Media Analysis, pages 161–169, San Diego, California. Association for Computational Linguistics.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in Twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 581–586, Portland, Oregon, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv: 1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *arXiv preprint arXiv: 1711.05101*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv: 1301.3781*.

Shabnam Nazmi, Xuyang Yan, Abdollah Homaifar, and Emily Doucette. 2020. Evolving multi-label classification rules by exploiting high-order label correlations. *Neurocomputing*, 417:176–186.

Bo Peng, Jin Wang, and Xuejie Zhang. 2018. YNU-HPCC at SemEval-2018 task 3: Ensemble neural network models for irony detection on Twitter. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 622–627, New Orleans, Louisiana. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2011. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460, Osaka, Japan. The COLING 2016 Organizing Committee.