PALI-NLP at SemEval-2022 Task 6: iSarcasmEval- Fine-tuning the Pre-trained Model for Detecting Intended Sarcasm

Xiyang Du and **Dou Hu** and **Meizhi Jin** and **Lianxin Jiang**
and **Yang Mo** and **Xiaofeng Shi**
Ping An Life Insurance Company of China, Ltd.
{DUXIYANG037, HUDOU470, JINMEIZHI005
JIANGLIANXIN769, MOYANG853, SHIXIAOFENG309}
@pingan.com.cn

Abstract

This paper describes the method we utilized in the SemEval-2022 Task 6 iSarcasmEval: Intended Sarcasm Detection In English and Arabic. Our system has achieved 1st in SubtaskB, which is to identify the categories of intended sarcasm. The proposed system integrates multiple BERT-based, RoBERTa-based and BERTweet-based models with finetuning. In this task, our contribution is listed as follow: 1) we reveal several large pre-trained models' performance on tasks coping with the tweet-like text. 2) Our methods prove that we can still achieve excellent results in this particular task without a complex classifier adopting some proper training method. 3) we found there is a hierarchical relationship of sarcasm types in this task.

1 Introduction

Generally speaking, when we communicate through natural language, the literal meaning of the words is consistent with the meaning we want to express. Sarcasm is a form of linguistic expression when this "congruence" is broken (Wilson, 2006).

Due to the inherent metaphorical nature and subtle sentimental expression of this particular form of language expression. The detection task related to this kind of text, which is a negative expression of a positive emotion or the positive expression of negative emotion, is extremely difficult for machines (Yaghoobian et al., 2021). This sarcasm data also weakens the detection modules that are widespread in our society (Maynard and Greenwood, 2014).

Previous work shows that sarcasm often comes with incongruity between expectation and reality (Gibbs Jr et al., 1994). Many works attempt to model this incongruity within the text (Tay et al., 2018; Xiong et al., 2019). For multi-model data, some works use the features from different modalities (Schifanella et al., 2016; Cai et al., 2019), and some works shows that inter-modality incongruity is also an important feature for multi-modal sarcasm detection (Pan et al., 2020).

The SemEval-2022 Task6 (Abu Farha et al., 2022) is designed to detect sarcasm and sarcasm types in twitter texts. In Subtask B, if a tweet is not sarcastic, it should not be annotated with any sarcasm label; if it is sarcastic, we need to detect which sarcasm it is, and it could have different sarcasm type at the same time. The main metric of this task is the Macro-F1 score of all sarcasm types.

We design a simple and effective system for this task. The system is based on a large-scale pre-trained model based on bi-directional transformers (Vaswani et al., 2017) and fine-tuned to obtain the final output. First, we augmented the iSarcasm dataset with additional datasets, and then we set an appropriate learning rate for each layer and set the model with an appropriate initialization state. Next, we strengthen the model's generalization ability through adversarial training, multi-sample dropout and other approaches. Finally, we use the [CLS] token of the last layer of the encoder to perform fine-tuning on the final training dataset and ensemble them using the hierarchical way.

2 System Overview

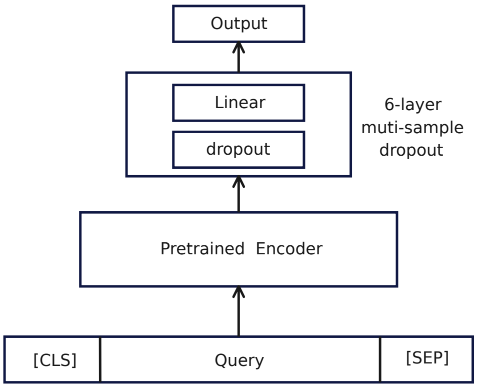

Figure 1: The overall architecture

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 815 - 819
July 14-15, 2022 ©2022 Association for Computational Linguistics

Figure 1 shows our model architecture. This task is a multi-label text classification task, so we follow the common input format of BERT, that is, using [CLS] and [SEP] as the starting and ending token of the text. In addition to this, we applied data cleaning to the text, replacing "@xxx" with <user>, "#xxx" with <tag>, and "http:xxx" with <url>. It is worth mentioning that we did not do any processing on emojis because we think emojis may be an important feature representing the gap between text semantics and underlying sentiment. These actions are done automatically by the tokenizer. The obtained final token embedding, segment embedding and position embedding constitute the input of the pre-trained model encoder.

In the last layer of the pre-trained encoder, we can acquire the representation of all tokens. In this task, we only select the representation of the first [CLS] token. After that, a layer-norm operation and multi-sample dropout will be utilized on the representation from the encoder. Finally, we use BEC loss as our loss function.

2.1 Pretrained Model

Our submitted architecture integrates three pre-trained language models of different architectures.

BERT-base(BERT) (Devlin et al., 2018): BERT adopts the multi-layer bidirectional transformer encoder to obtain the representation of a query. In the pre-training procedure, BERT conducts two different pre-training objectives. 1. Masked language modelling (MLM) objective. This task predicts a masked token based on a randomly masked input. 2. Next sentence prediction (NSP) objective. The goal of this task is to predict whether the second sentence is the following sentence of the first one.

RoBERTa-base(RoBERTa) (Liu et al., 2019): RoBERTa adopts the same model architecture as BERT and improves the pre-training. It believes that the pre-training of BERT is insufficient, so RoBERTa executes pre-training using longer sentences, more data, and a larger batch size than BERT uses. The author also believes that the NSP task of BERT is redundant and removed NSP from pre-training. Meanwhile, the MLM task is improved at the same time, and the token is dynamically masked during the training process.

BERTweet-base(BERTweet) (Nguyen et al., 2020): BERTweet follows the RoBERTa training procedure, and it is the first large-scale pre-trained language model that uses Twitter texts as a pre-training corpus. Therefore, BERTweet has better performance on tasks related to the tweet-like text.

2.2 Adversiral Training

We also incorporate adversarial training into the training process. The objective of adversarial training is to improve the generalization of the model by perturbing the embedding. For the calculation of this perturbation, we mainly implement two different methods. The Fast Gradient Method (FGM) calculates the disturbance at the moment through the gradient (Miyato et al., 2016), while the Projected Gradient Descent (PGD) executes this process through more steps and additionally adds spherical mapping to prevent the perturbation from being too large (Madry et al., 2017). During the training process, we adopt adversarial training on both the embedding layer and the first layer of the encoder.

2.3 Multi-sample Dropout

Dropout is a common and effective way to increase the generalization of deep neural networks. It can effectively reduce the overfitting of the model by ignoring some neurons in training according to a certain probability. The multi-sample dropout (Inoue, 2019) we use in this paper is an enhanced dropout method. It goes through multiple dropout operations and averages the output obtained by each dropout operation as the final output. In multi-sample dropout, the weights of each dropout layer and classifier layer will be shared, so multi-sample dropout can achieve better results than the original dropout and not bring a significant increase in computational cost.

2.4 Contrastive Loss

Contrastive learning has drawn attention for its' excellent performance. The main idea is to shorten the distance between similar samples(in this task, similar means having the same label) and separate the samples that are not similar. In the SubtaskB, we mainly implement supervised contrastive loss (SupConLoss)(Khosla et al., 2020).

2.5 Ensemble method

For this task, we adopt a hierarchical model ensemble approach. First, we give the models corresponding voting weights based on the performance of each model on the validation set. The weights are calculated as the square root of the inverse of the model's rank among all models. Then we conduct two votes, one for the first two labels (Sarcasm, Irony) and one for the last four labels (Satire, understatement, overstatement, rhetorical question). In this way, we get the final output.

3 Experimental Setup

3.1 Dataset

iSarcasm-2022. Dataset of SemEval-Task6, consisting of tweets text and correspoding sacarsm types. For tweets that are sarcasm, the sarcasm type is given as annotated by language experts, and a single tweet may contain one or more sarcasm types. Along with the sarcastic tweets is the "Rephrase" written by the same poster of the tweet. "Rephrase" contains the same expressive meaning as sarcastic tweets but without sarcasm.

iSarcasm (Oprea and Magdy, 2019). iSarcasm dataset, consisting of tweets texts and correspoding sacarsm types just like iSarcasm-2022. We only leverage the sarcastic tweet which is identityfied as sarcasm. The publisher of sarcastic tweets provides the sarcasm type of iSacarsm. No "Rephrase" is provided.

3.2 Training Details

We fine-tune the pre-train models with batch size 128, sequence length 64,multi-sample dropout of 0.4, threshold 0.2. We set peak learning rate 1e5 for ten epochs and apply layer-wise learning rate Decay for each layer. We set AdamW and Lookahead as our optimizer and set cosine warm-up during the first 0.1 of the updates followed by a linear decay. We conduct validation three times per epoch and perform early stopping. We set the multi-sample layer to six layers and adopt PGD on the embedding and first encoder layers. The training is done on NVidia V100 GPUs. All the F1 result is performance on the test set.

4 Results

Our model perfomence is shown in Table 1

4.1 Pretrained Model Selection and Data Analysis

We try three transformer-based pre-trained language models in this task, BERT, RoBERTa and BERTweet. From the Table 1 we can see that although RoBERTa achieves better results than BERT, they both perform far worse than BERTweet. The possible reason is due to the difference in the pre-training corpus. The dataset texts of this competition are all tweeted texts of users, and these texts have linguistic features like hashtags and emoji, making them significantly different from standard texts. BERTweet conducts pre-training on tweet texts while BERT and RoBERTa do not have such settings, which leads to BERTweet being better able to handle this particular type of data.

For the fine-tune dataset selection, we extend the dataset provided by the organiser with an additional 777 samples based on the competition requirements that additional data could be used. By analysing the dataset, we found that: 1) The competition dataset reflects a long tail that Understatement, Overstatement, and Rhetorical question are pretty rare. 2) There is an apparent hierarchical relationship between each label. We analysed the combination of labels and found that six labels can be categorised into primary labels (sarcasm, irony) and secondary labels(satire, understatement, Overstatement, rhetorical question). The hierarchical relationship is presented in the Fig 2 . The standard BEC loss and re-weighted BEC loss are tested in this task. The result is shown in the Table 2. Models gain a significantly 10.37% increase in the test set from 0.1417 to 0.1564 of macro-f1 score in the latter setting. We used this system as our baseline.

4.2 Multi-sample Dropout

We experiment with four different multi-sample dropout layer settings, ranging from 2 to 8 layers and the result is shown in Tabel 3. We finally implement 6-layer multi-sample dropout in this task, which achieved 0.1578 in macro-f1 compared to the 0.1564 of baseline.

4.3 Adversiral Training

Tabel 4 display the performance of different adversarial training strategies. We conduct experiments on two common used adversarial training methods and test the effect of the adversarial rate. In the end, we find that PGD is slightly better than FGM as a whole. When the ratio is 0.5, PGD is optimal as adversarial rate 0.5, which increases 1.98% compared to the baseline. If combined with the optimal multi-sample dropout method, the model can obtain 7.10% improvement, reaching a macro-f1 score of 0.1675.

4.4 Contrastive Loss

We experiment with SupCon loss. We can see the result on Tabel 5. There is an indeed decrease when

Model	Macro-F1	F1-SCM	F1-IRN	F1-ST	F1-UST	F1-OST	F1-RQ
BERT-base-uncased	0.0766	0.2605	0.0976	0.0000	0.0000	0.0000	0.1013
RoBERTa-base	0.0991	0.2921	0.1915	0.0000	0.0000	0.0000	0.1111
BERTtweet-base	0.1417	0.4749	0.2151	0.0000	0.0000	0.0000	0.1600
Our Baseline	0.1564	0.4760	0.1630	0.0667	0.0000	0.0976	0.1441
Our Submitted Model	0.1630	0.4828	0.1863	0.0667	0.0000	0.0870	0.1556
Our Best Model(single)	0.1675	0.4586	0.1854	0.1000	0.0000	0.0930	0.1682

Table 1: Performance of final result

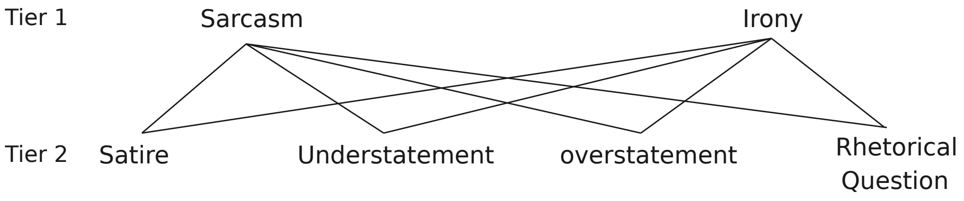

Figure 2: The hirechical relationship of sarcasm types

Model(base)	Loss	Macro-F1
RoBERTa	non-weighted	0.0991
RoBERTa	re-weighted	0.1034
BERTweet	non-weighted	0.1417
BERTweet(Base)	re-weighted	0.1564

Table 2: Performance of different pre-trained models applying non-weighted or re-weighted loss

Setting	Macro-F1
Base	0.1564
Base+M-dropout	0.1578

Table 3: Performance of models applying multi-sample dropout

Setting	M-dropout	Ad-rate	Macro-F1
Base	False	0	0.1564
Base+PGD	False	0.5	0.1595
Base+FGM	False	0.5	0.1593
Base	True	0	0.1578
Base+PGD	True	0.5	0.1675

Table 4: Performance of models applying different adversarial training strategies

Setting	M-dropout	Macro-F1
Base	False	0.1564
Base+SupCon	False	0.1529
Base	True	0.1578
Base+SupCon	True	0.1670

Table 5: Performance of models applying SupCon loss

performing SupCon loss. However, when multi-sample dropout is adopted along with SupCon loss, it shows excellent results, an increase of 6.77% compared to baseline. See Table 5 for performance of contrastive loss applied.

5 Conclusion

We employ the large pre-trained models and fine-tune them for sarcasm category discrimination. We compare the performance of different pre-trained models on Subtask B of SemEval-2022 Task 6. The results show that the difference between the pre-training corpus and the downstream task corpus will significantly affect the performance of the model. We find that the pre-trained model using the default training settings performed poorly on this task, and good model initialization and training strategies can help improve this situation. We also find that there is a hierarchical relationship between the types of sarcasm which we believe is an important feature worth exploiting.

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.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Raymond W Gibbs Jr, Raymond W Gibbs, and Jr Gibbs. 1994. *The poetics of mind: Figurative thought, language, and understanding*. Cambridge University Press.

Hiroshi Inoue. 2019. Multi-sample dropout for accelerated training and better generalization. *arXiv preprint arXiv:1905.09788*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673.

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*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Diana G Maynard and Mark A Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Lrec 2014 proceedings*. ELRA.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*.

Silviu Oprea and Walid Magdy. 2019. isarcasm: A dataset of intended sarcasm. *arXiv preprint arXiv:1911.03123*.

Hongliang Pan, Zheng Lin, Peng Fu, Yatao Qi, and Weiping Wang. 2020. Modeling intra and inter-modality incongruity for multi-modal sarcasm detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1383–1392.

Rossano Schifanella, Paloma de Juan, Joel Tetreault, and Liangliang Cao. 2016. Detecting sarcasm in multimodal social platforms. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1136–1145.

Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. *arXiv preprint arXiv:1805.02856*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Deirdre Wilson. 2006. The pragmatics of verbal irony: Echo or pretence? *Lingua*, 116(10):1722–1743.

Tao Xiong, Peiran Zhang, Hongbo Zhu, and Yihui Yang. 2019. Sarcasm detection with self-matching networks and low-rank bilinear pooling. In *The World Wide Web Conference*, pages 2115–2124.

Hamed Yaghoobian, Hamid R Arabnia, and Khaled Rasheed. 2021. Sarcasm detection: A comparative study. *arXiv preprint arXiv:2107.02276*.