Tsia at SemEval-2021 Task 7: Detecting and Rating Humor and Offense

Zhengyi Guan, Xiaobing Zhou*
School of Information Science and Engineering Yunnan University, Yunnan, P.R. China
*Corresponding author:zhouxb@ynu.edu.com

Abstract

This paper describes our contribution to SemEval-2021 Task 7:Detecting and Rating Humor and Offense.This task contains two sub-tasks, sub-task 1 and sub-task 2. Among them, sub-task 1 contains three sub-tasks, sub-task 1a ,sub-task 1b and sub-task 1c.Sub-task 1a is to predict if the text would be considered humorous.Sub-task 1c is described as follows: if the text is classed as humorous, predict if the humor rating would be considered controversial, i.e. the variance of the rating between annotators is higher than the median.we combined three pre-trained model with CNN to complete these two classification sub-tasks.Sub-task 1b is to judge the degree of humor.Sub-task 2 aims to predict how offensive a text would be with values between 0 and 5.We use the idea of regression to deal with these two sub-tasks.We analyze the performance of our method and demonstrate the contribution of each component of our architecture.We have achieved good results under the combination of multiple pre-training models and optimization methods.

1 Introduction

Humor is an intellectual activity that can cause certain emotions in human thinking. Humor is not only very important but also very common in daily life. People's research on humor has involved many fields such as psychology, sociology, linguistics and so on.Of course, it also has special value for the research of computing languages.Because of its complexity and inherent subjectivity, the development of automatic humor recognition and assessment poses a great challenge in Computational Linguistics, and therefore is a popular subject in various shared task competitions.(Dick et al., 2020)However, we must also recognize the difficulties of humor research.First of all, although humans can easily judge whether a sentence is humorous in daily life, but due to humor is restricted by geography, environment, social background and other aspects, we usually not only pay attention to this sentence or whether the matter is humorous, We have to figure out how humorous or funny this context is? In other words, we pay more attention to its degree of humor which is not so easy for computers.And because humor is affected by the environment, different people have different understanding of humor. Just like sometimes your humor is based on the suffering of others. Things you find funny, but others don't necessarily find them funny. In other words, humor is controversial. So we have to determine the specific humor rate. This task is to take a median as the criterion for humor. It is also difficult to judge the humor of this dispute by computer language.

More recently, some humorous sentences can also have derogatory and offensive elements. Whether humor can cause offense is also one of the researches in this thesis. I believe that the study of humor not only helps to improve the computer's understanding of humor in certain aspects, but also purifies our network environment.

The four sub-tasks of SemEval-2021 task7 are designed to solve the above problems. For deep learning, the computer must not only judge whether a sentence is humorous.It is more important to understand this humorous sentence.In our paper, we designed two effective systems to solve the above four sub-tasks. For Sub-task 1a and Sub-task 1c, We take them as a binary classification task.we designed an efficient system using the idea of BERT-CNN.This is not a new idea because people have tried in the past.We also use the other popular pre-training models,including the derived ALBERT and RoBERTa.For sub-task 1a, we need to judge whether a sentence is humorous.We predict a Label L: where L $\in \{is\_humor - 1, not\ is\_humor - 0\}$.For sub-task 1c, we also need to judge whether a sentence is controversial and predict a Label L: where L$\in$ $\{humor\_controversy - 1, not\ humor\_controversy - 0\}$.For sub-task 1b,

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1108–1113
Bangkok, Thailand (online), August 5–6, 2021.

we combined regression ideas with the current popular pre-training model to complete these two subtasks.The two input sentences were split into two lists and fed into the Regression Model, which made a prediction about the funniness of each sentence.Then we compared the results of the prediction to determine the funnier of the two sentences.We compare the humor between each sentence and finally return a humor value. (Ammer and Grüner, 2020)Finally, the humor rate and the level of controversy are mapped to the range from 0 to 5.We use mean square error to measure these two tasks.

2 Background

The judgment of humor is the same as other text classification problems in natural language processing.The most important thing is to find suitable features to represent sentences.The task is to assign predefined categories to a given text sequence. many work has shown that pre-trained models on large corpora are beneficial for text classification and other NLP tasks, which can avoid training new models from scratch.Since 2013,people have proposed some word embedding approaches such as word2vec (Mikolov et al., 2013) and glove (Pennington et al., 2014).However,because their word embeddings are all in the same space, they cannot express the role of polysemy.In other words,they are non-contextual embedding, they cannot capture the high-level concepts of sentences, such as semantics and context (Sun et al., 2020).Later, someone proposed the ELMo model to solve this problem.Compared with word2vec and glove, ELMo captures contextual information and not just individual information of words. In word2vec, the vector representations of words are completely consistent in different contexts, and ELMo is optimized for this (Zhang et al., 2017). More recently, pre-trained language models have shown to be useful in learning common language representations by utilizing a large amount of unlabeled data: such as OpenAI GPT (Brown et al., 2020) and BERT (Devlin et al., 2018). BERT is based on a multi-layer bidirectional Transformer (Vaswani et al., 2017) and is trained on plain text for masked word prediction and next sentence prediction tasks.This paper tried other two new pre-training models of ALBERT (Lan et al., 2019) and RoBERTa (Liu et al., 2019) based on BERT. And we fine-tuned downstream tasks for a variety of pre-trained models. Finally,We completed these four sub-tasks effectively.

3 System overview

3.1 Data

The data of the four sub-tasks are all provided by SemEval(Meaney et al., 2021).The official organizer provides the same training set and test set for all sub-tasks. We split the training set into a new training set and a test set by using the stratified 5-fold cross-validation.BERT uses the wordpiece tool for word segmentation and inserts special separators($[CLS]$which are used to separate each sample) and separator($[SEP]$ which are used to separate different sentences in the sample). For each fold of the data set, the input data format is as follows: $[CLS]$+ $[sentence]$+ $[SEP]$(Bai and Zhou, 2020).There are a total of 8000 data in the training set and 1000 data in the test set.In addition,the training set was split into 85% and 15% for training and development set respectively(Note:We did not include the use of the development dataset which was given by the task organizers).They are all English sentences. Each sentence of the data in the training set can be regarded as a combination of $\{$*id,text*$\}$ and one of $\{$*is_humor,humor_rating, humor_controversy,offense_rating*$\}$.For the data we did a simple pre-processing.We first remove the characters specified at the beginning and end of the string.(the default is a space or a newline.)

3.2 Methodology

Text classification technology is an efficient information retrieval and data mining information technology.The classification method based on machine learning trains a classification model by learning a given training set, and then uses the training model to classify text. Traditional machine learning classification methods include: random forest (RF), naive Bayes (NB), logic Regression (LR) and Support Vector Machine (SVM), etc.However, with the development of deep learning, many NLP tasks can adopt a pre-training + fine-tuning structure.The most typical is the BERT pre-training model.We propose two architectures to solve four sub-tasks.

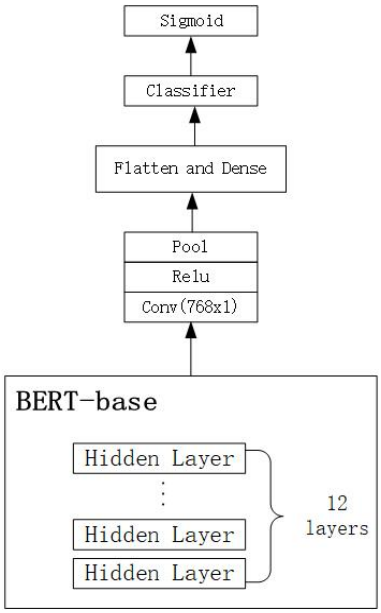

Figure 1: Model for Sub-task 1a and Sub-task 1c

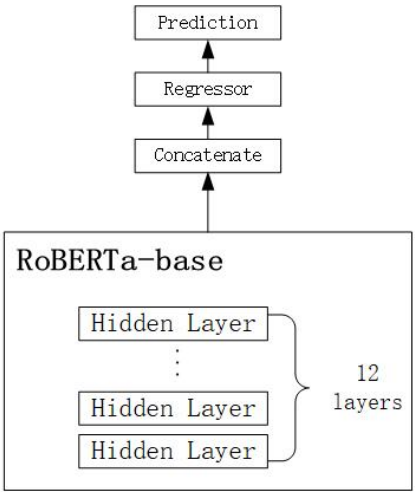

Figure 2: Model for Sub-task 1b and Sub-task 2

3.2.1 Method A

Method A is designed to solve sub-task 1a and sub-task 1c.CNN for textual tasks by Kim (Kim, 2014) showed superiority in text classification tasks. CNN can be used with learned vector representations of the text (embeddings). These embeddings may either be initialized randomly and trained along with the model, or can be pre-trained vectors.

The proposed model maximizes the utilization of knowledge embedded in pre-trained BERT language models by feeding the outputted contextualized embeddings of its last four hidden layers into a several filters and convolution layers of the CNN. Finally, the output of the CNN was passed to a dense layer and the predictions were obtaineds(Safaya et al., 2020).

As shown in Figure 1, we use BERT-base as a pre-training model to build the model and other pre-training models are similar. BERT is a model built based on Transformer Encoder. Its entire architecture is actually based on DAE (Denoising Autoencoder).This part is called Masked Lanauge Model (MLM) in the BERT (Devlin et al., 2018) article. MLM is not strictly a language model, because the entire training process is not trained using a language model. BERT randomly replaces some words with the MASK tag, and then predicts the word masked. The process is actually the process of DAE. BERT has two main trained models, namely BERT-Small and BERT-large. BERT-large uses a 12-layer encoder structure, that is, twelve hidden layers. The whole model has a lot of parameters. For sub-task 1a and sub-task 1c, we tried a variety of methods based on BERT-base, including BERT+LSTM and other pre-trained models ALBERT and RoBERTa(add a linear layer). The last method to try is after BERT pre-training model, we use one or two layers of CNN to perform feature extraction. Finally, we input into a linear classifier to classify English sentences (humorous or controversial).

3.2.2 Method B

Method B is used to solve sub-task 1b and subtask 2(Regression tasks). Since the values we want to output (values between 0 and 5) are continuous.We pre-trained through the input of two English sentences and then made a humorous (controversial) comparison. Since the effect of CNN on the regression task is not very useful, we mainly tried and improved on the pre-training model.We mainly use BERT, ALBERT and RoBERTa for word embedding. as shown in Figure 2. RoBERTa works best. BERT has the worst effect. Because RoBERTa is trained with dynamic masking, FULL SENTENCES without NSP loss,large mini-batches and a larger byte-level BPE(Liu et al., 2019).In addition, it adjusted the parameters of the Adam algorithm. From 16G data to 160G. ROBERTa uses a larger batch size, and the number of training is more. The network structure is complex, so the fitting effect is better. Finally, we throw the trained model into a regression model to calculate the humor rate and controversial rate.

4 Experimental setup

The code this time is mainly based on Transformers under Hugging Face. The neural network tool we use is PyTorch.For sub-task 1a and sub-task 1c,we use the same method A(the same structure).We just read different id of training set.Sub-task 1b and sub-task 2 used the other method B.

4.1 Hyper-parameters

In this work, because our models are implemented based on PyTorch. We use the BERT-base+CNN as our sub-task1a and sub-task 1c's pre-trained model. For all models, in order to save GPU memory, the batch size parameter of GPU in fine-tuning is set to 8 and the gradient accumulation steps(gas) is set to 1, so that each time a sample is an input,the gradient is accumulated 1 times, and then the back-propagation update parameters are performed. The memory is saved by sacrificing a certain training speed;learning rate is 5e-5. we use the triangular learning rate. First, the learning rate is gradually increased through warm up, and then the linear learning rate is gradually reduced through linear learn rete decay, which effectively improves the training effect.(Bai and Zhou, 2020) The hyper-parameters of each model and the results on the test set are shown in Table 1.

Tasks	Hyperparameters
Sub-task 1a and sub-task 1c	lr=5e-5 output hidden states=True epoches=5 per gpu train batch size=8 gas=1 filt size=(3,4,5) num filter=(3,4,5) hidden size=68 dropout=0.2
Sub-task 1b and sub-task 2	output hidden states =True dropout=0.1 lr=5e-5 epochs=10 per gpu train batch size=8 gas=1

Table 1: Hyperparameters of the used model

4.2 Prediction module

For sub-task 1a and sub-task 1c, we mainly use Precision and F1-score to evaluate our model A.The criteria evaluation of F1-score is as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{Precision * Recall * 2}{Precision + Recall}$$

For task 1b and task 2,we use RMSE to evaluate our model. RMSE is as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}}$$

5 Results

For sub-task 1a and sub-task 1c, we first tried BERT-base as our word embedding model. On this basis, we added LSTM (Wang et al., 2018) to further extract the features of words. Long-term short-term memory (LSTM) network has the ability to maintain long-term memory. The ability of has proven to be particularly useful for learning sequences containing long-term patterns of unknown length.We also tried two other pre-training models (ALBERT and RoBERTa) as a comparative experiment. These two new language models have made some improvements on the basis of BERT, but they have different effects on different data sets. RoBERTa is suitable for complex neural network architecture, ALBERT architecture is relatively streamlined. From the data in Table 2 , it can be seen that they are almost the same as BERT in extracting word features.Finally, we chose to add CNN on the basis of BERT to extract the features of words, and found that the effect is better than LSTM and other pre-training models.

Since CNN is not very effective in dealing with regression problems, we mainly use RoBERTa as our system architecture.From Table 3, we can find that compared to BERT, RoBERTa's RMSE is far better than BERT-base under the same training epochs.

6 Conclusion

In this paper,we gave a description of the BERT+CNN architecture and the popular RoBERTa pre-training model architecture, and finally solved four sub-tasks. The best F1 for Sub-task 1a is 0.9206 and the best F1 for Sub-task

1c is 0.6744. The best root mean square errors of Sub-task 1b and Sub-task 2 are 0.6510 and 0.5588. Our four sub-tasks all appeared on the leaderboard(Note: The data from our computer is a little different from the official evaluation results.The results of the final four sub-tasks in the leadboard are shown in Table 4.).Experiments have shown that CNN has a certain effect on text classification, but this time only one layer of CNN was added to the Classifier. In addition, the experiment also shows that RoBERTa has a better effect than BERT in dealing with regression problems.In the future, we will try to integrate and distill the model and process the data.we consider to introduce external knowledge to model headlines and improve the humor recognition performance.

Method	Task 1a		Task 1c	
	F1	Acc	F1	Acc
BERT	0.9175	0.9310	0.6659	0.6835
ALBERT	0.9162	0.9210	0.6576	0.6911
RoBERTa	0.9176	0.9320	0.6488	0.7300
BERT+LSTM	0.9054	0.9120	0.6519	0.6806
BERT+CNN	**0.9206**	0.9250	**0.6744**	0.7025

Table 2: the Table for Sub-task 1a and Sub-task 1c

Method	Sub-task 1b	Sub-task 2
	RMSE	RMSE
BERT	1.7161	1.826
ALBERT	0.6700	0.6576
RoBERTa	**0.6510**	**0.5588**

Table 3: the Table for Sub-task 1b and Sub-task 2

Task	Best Result
sub-task 1a	0.9205(F1)
sub-task 1b	0.7010(RMSE)
sub-task 1c	0.4271(F1)
sub-task 2	0.5419(RMSE)

Table 4: Final Result on the Leaderboard

7 Acknowledgments

Our work was supported by the Natural Science Foundations of China under Grants 61463050, the NSF of Yunnan Province under Grant 2015FB113.

References

Charlotte Ammer and Lea Grüner. 2020. UniTuebingenCL at SemEval-2020 task 7: Humor detection in news headlines. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1060–1065, Barcelona (online). International Committee for Computational Linguistics.

Yang Bai and Xiaobing Zhou. 2020. BYteam at SemEval-2020 task 5: Detecting counterfactual statements with BERT and ensembles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 640–644, Barcelona (online). International Committee for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, and Dario Amodei. 2020. Language models are few-shot learners.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Anna-Katharina Dick, Charlotte Weirich, and Alla Kutkina. 2020. HumorAAC at SemEval-2020 task 7: Assessing the funniness of edited news headlines through regression and trump mentions. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1019–1025, Barcelona (online). International Committee for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

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. *CoRR*, abs/1907.11692.

J.A. Meaney, Steven R. Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. Semeval 2021 task 7: Hahackathon, detecting and rating humor and offense. In *Proceedings of the 59th*

Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Computer Science*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune BERT for text classification?

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Jin Wang, Bo Peng, and Xuejie Zhang. 2018. Using a stacked residual LSTM model for sentiment intensity prediction. *Neurocomputing*, 322(DEC.17):93–101.

Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, and Lawrence Carin. 2017. Deconvolutional paragraph representation learning. In *NIPS (2017)*.