DeepBlueAI at SemEval-2021 Task 7: Detecting and Rating Humor and Offense with Stacking Diverse Language Model-Based Methods

Bingyan Song **Chunguang Pan** **Shengguang Wang** **Zhipeng Luo**
DeepBlue Technology (Shanghai) Co., Ltd
{songby, panchg, wangshg, luozp}@deepblueai.com

Abstract

This paper describes the winning system for SemEval-2021 Task 7: Detecting and Rating Humor and Offense. Our strategy is stacking diverse pre-trained language models (PLMs) such as RoBERTa and ALBERT. We first perform fine-tuning on these two PLMs with various hyperparameters and different training strategies. Then a valid stacking mechanism is applied on top of the fine-tuned PLMs to get the final prediction. Experimental results on the dataset released by the organizer of the task show the validity of our method and we win first place and third place for subtask 2 and 1a.

1 Introduction

Humor and offense detection continue to be challenging AI problems since humor and offense involve in-depth world-knowledge, common sense, and the ability to perceive relationships across entities and objects at various levels of understanding (Hossain et al., 2019). The recognition of humor and offense in the text has been receiving much attention (Zampieri et al., 2019; Hossain et al., 2020). Accordingly, SemEval-2021 Task 7, **Detecting and Rating Humor and Offense**, which aims to automatically recognize humor in English jokes was held (Meaney et al., 2021).

In this paper, we introduce our system for accomplishing the above task by leveraging pre-trained models (PLMs). There are two main steps for our system, i) fine-tuning two kinds of PLMs, including ALBERT (Lan et al., 2019) and RoBERTa (Liu et al., 2019) with various hyperparameters and training strategies, achieving diverse models; ii) applying a validity stacking mechanism on top of these PLMs to do the final predictions.

Our experimental results show that merging PLMs with different training strategies together can achieve great improvement which verifies the effectiveness of increasing model diversity. As a result, our system achieves the F1-score of 96.76% in subtask 1a and the RMSE of 41.2% in subtask 2, which ranks third and first among all the participated teams respectively.

Tags	No. of is_ humor	Percentage
train	4932	61.65%
dev	632	63.20%
test	615	61.50%

Table 1: The number and percentage of humor samples in training set, validation set and test set respectively.

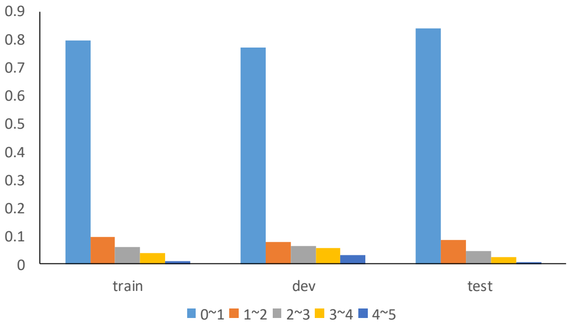

Figure 1: The distributions of offense rating for training set, validation set and test set.

2 Background

2.1 Task Definition

The "Detecting and Rating Humor and Offense" task, shared by SemEval-2021, consists of two subtasks. Subtask 1 includes three parts, a) A binary task to predict if the text would be considered humorous; b) A regression task to predict how humorous a text is if it is classed as humorous; c) A binary task to predict if the humor rating would be considered controversial when the text is classed as humorous. Subtask 2 aims to predict how offensive a text would be with values between 0 and 5. This score can be calculated regardless of whether the text is classed as humorous or not. In this paper, we mainly focus on subtask 1a and subtask 2.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1130–1134
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

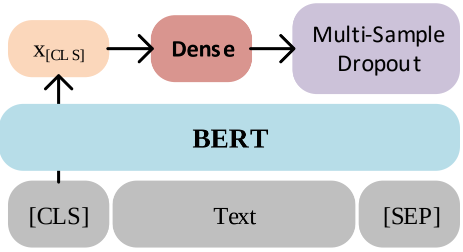

Figure 2: The overall architecture for detecting and rating humor and offense.

2.2 Dataset

Humor and Offense appreciation is a highly subjective phenomenon, with age, gender, race, and socioeconomic status are known to have an impact on the perception of a joke. The labels and ratings of the English dataset in this task are collected from a balanced set of age groups from 18-70 and are various in genders, political stances, and income levels. The dataset has a total of 10,000 samples, which are divided into training set, validation set, and test set according to 8:1:1. Table 1 shows the number and percentage of humor samples in the three datasets and we can find that their distributions are very similar with nearly 60% are humor ones. Figure 1 demonstrates the distribution of offense ratings in three datasets. Samples with offense ratings between (0,1) are the most and the three datasets have the same distribution of offense ratings as well.

3 System Overview

3.1 PLMs-based Method

Architecture In our method, we have the same architecture for dealing with subtask 1a and subtask 2. As shown in Figure 2, we utilize several pre-trained language models (e.g., RoBERTa) as the encoder and segment different texts with special tokens `[CLS]` and `[SEP]`. After the tokenization, we can get the embedding of `[CLS]`, which can be seen as the representation for the whole input text. We pass it through a dense layer and obtain the final prediction through the Multi-Sample Dropout (Inoue, 2019). The output of dense layer x is depicted as below,

$$x = ReLU(W_0 dropout(x_{[CLS]}) \qquad (1)$$

where $W_0 \in R^{d \times k}$ is the learning weight, k is the dimension of $e_{[CLS]}$ and d is a hyperparameter which we set as 256 and the dropout rate here we set as 0.2 or 0.5.

Multi-Sample Dropout Dropout is a simple but efficient regularization technique for achieving better generalization of deep neural networks. During training, dropout randomly discards a portion of the neurons to avoid overfitting. The original dropout creates a randomly selected subset (called a dropout sample) from the input in each training iteration while the multi-sample dropout creates multiple dropout samples. The loss is calculated for each sample, and the sample losses are averaged to obtain the final loss.

Thus, the final prediction of both subtask 1a and 2 can be computed as follows,

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} Sigmoid(W_i dropout_i(x)) \qquad (2)$$

where $W_i \in R^{1 \times d}$ is the learning weights, N is the number of dropout values which we set as 5. By using this training mechanism, we can accelerate training and achieve lower error rates as well. Since the Sigmoid function used here is the logistic function which maps any real value to the range (0,1), we preprocess the rating of offense in subtask 2 from (0,5) to (0,1).

Loss function As mentioned above, subtask 1a is a binary task and subtask 2 is a regression task, thus we choose Binary Cross Entropy (BCE) and Mean Square Error (MSE) as the loss function respectively.

3.2 Training strategies

To further improve the diversity and accuracy of trained models, we incorporate three training strategies as depicted below.

Task-Adaptive Pre-training Task-adaptive pre-training (TAPT) is an effective method to improve model performance (Gururangan et al., 2020). The data used in general pre-training usually vary from task-specific data. Thus we do task-adaptive by pre-training the masked language model task on the given Humor and Offense dataset.

Pseudo-Labelling Pseudo labeling (PL) is the process of using a labeled data model to predict labels for unlabeled data. We predict the unlabeled test dataset and mix these pseudo labels with the training set together to train the new model. For

Subtask 1a		Subtask 2	
Model	**F1**	**Model**	**RMSE**
ALBERTBASE	0.9635	-	-
ALBERTBASE+AT	0.9662	-	-
RoBERTaLARGE	0.9685	RoBERTaLARGE	0.4846
RoBERTaLARGE+AT	0.9694	RoBERTaLARGE+AT	0.4713
RoBERTaLARGE+TAPT	0.9724	RoBERTaLARGE+TPAT	0.4621
RoBERTaLARGE+TAPT+AT	0.9727	RoBERTaLARGE+TAPT+AT	0.4607
RoBERTaLARGE+TAPT+KD	0.9714	RoBERTaLARGE+TAPT+KD	0.4633
RoBERTaLARGE+TAPT+KD+AT	0.9726	RoBERTaLARGE+TAPT+KD+AT	0.4605
RoBERTaLARGE+TAPT+PL	0.9728	RoBERTaLARGE+TAPT+PL	0.4571
RoBERTaLARGE+TAPT+PL+AT	**0.9738**	**RoBERTaLARGE+TAPT+PL+AT**	**0.456**

Table 2: Comparison of pre-trained language models with different training strategies of Subtask 1a and 2.

subtask 1a, we set the threshold as 0.8 which means samples with predicted scores higher than 0.8 are treated as the humor ones.

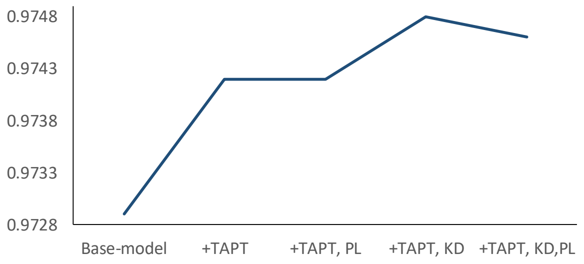

Figure 3: The comparison of F1 scores for stacking different models in subtask 1a.

Knowledge Distillation Inspired by (Hinton et al., 2015), we adopt the knowledge distillation (KD) mechanism into our system. The whole procedure consists of three steps. First, we train the original big model using a hard target, which is the true label given in the dataset. Next, we use the trained model to predict the soft target, which is the probability for each sample being humorous and offense. After this, we train a small model by minimizing the loss between the scores predicted by the small model and the soft target. The loss functions are still BCE and MSE. At last, we use the small model to predict the final results.

Adversarial Training Adversarial training (AT) is a popular approach to increasing the robustness of neural networks and has good regularization performance (Miyato et al., 2016). By adding perturbations to the embedding layer, we can get more stable word representations and a more generalized model, which significantly improves model performance on unseen data.

3.3 Stacking Trained Models

Model stacking is an efficient ensemble method to improve model accuracy. The main procedure of stacking trained models in our method including five steps. First, we use two different PLMs including RoBERTa and ALBERT. Second, we do TAPT on these PLMs to achieve new pre-trained models. Third, we perform 7-fold cross-validation on the whole training process to avoid overfitting or selection bias. Fourth, we train various models with different hyperparameters and different training strategies to improve the model diversity. Ultimately, we average all the predictions from different models to get the final prediction.

4 Experiments

Evaluation Metrics As mentioned in the official evaluation procedure of SemEval-2021 task 7, the main evaluation metrics for the binary classification tasks is f1-measure and the metric for the regression tasks is Root Mean Squared Error (RMSE).

Parameter settings All models are implemented based on the open-source transformers library of hugging face (Wolf et al., 2020), which provides thousands of pre-trained models that can be quickly downloaded and fine-tuned on specific tasks. To do better performance estimation, We gather the training set and validation set together as the new training set and then do 7-fold cross-validation on it. We set batch size as 16 and run 10 epochs for each fold. The learning rate is 1e-5. For RoBERTaLARGE and ALBERTBASE, the k is set as 1024 and 128 respectively.

5 Results

5.1 Ablation Studies

PLMs with Training Strategies For subtask 1a, we use two types of PLMs including ALBERTBASE and RoBERTaLARGE. As shown in Table 2. We set five groups of models and each group is the same models with or without adversarial training (AD).

The models of the first and second groups are the base ones and we add training strategies including task-adapative pre-training (TAPT), knowledge distillation (KD), and pseudo-labeling (PL) to the other three groups.

The results are the average scores from models with different hyperparameters (e.g. different dropout) by doing 7-fold cross-validation on the new training dataset depicted above. Since RoBERTa$_{\text{LARGE}}$ performs better on this task, ALBERT is not used in subtask 2 anymore. From Table 2, we find that for both subtask 1a and subtask 2, all the training strategies can improve the performance. Besides, models with AD achieve better scores than the ones without AD. The models adding TAPT, PL and AT together are the best ones.

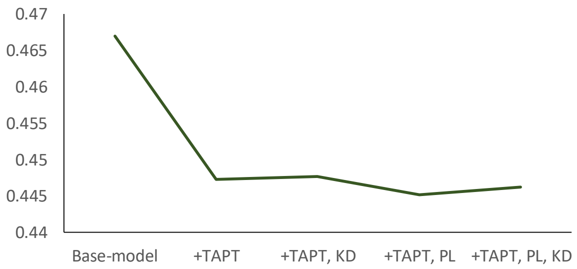

Figure 4: The comparison of RMSE for stacking different models in subtask 2.

Stacking trained models To stack the trained models, we use a simple method which averaging predictions from different models. Figure 3 and 4 show the comparison for stacking different models of subtask 1a and 2. We find that all scores of the ensemble ones are better than the best score in Table 2 which from a single model. This verifies the effectiveness of stacking different models.

However, both Figure 3 and 4 demonstrate that the best score is not to stacking models of all the groups in Table 2 but to stack part of the models. This indicates that combining the least correlated results is more efficient than combining them all.

Subtask 1a		**Subtask 2**	
System	**F1**	**System**	**RMSE**
endworld	0.9854	**DeepBlueAI**	**0.412**
stce	0.9797	mmmm	0.419
DeepBlueAI	**0.9676**	calamity link	0.423
baseline	0.9283	baseline	0.5770

Table 3: Leaderboard

5.2 Official Ranking

We submitted the scores predicted by the ensemble method introduced above. The official ranking is presented in Table 3. We rank third in subtask 1a and first in subtask 2, which verifies the validity of our system.

6 Conclusion

In this paper, we propose a top-performing approach for the task of Detecting and Rating Humor and Offense. We fine-tune two kinds of pre-trained language models including ALBERT and RoBERTa with different training strategies such as pseudo labeling and knowledge distillation. Then, we stack them with a simple linear regression model. Experimental results show the effectiveness of this ensemble method and we win first place and third place for subtask 2 and 1a. For future work, it would be interesting to test the performance of our best-performing system on other humor detection datasets to validate its portability and robustness.

References

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Nabil Hossain, John Krumm, and Michael Gamon. 2019. "president vows to cut¡ taxes¿ hair": Dataset and analysis of creative text editing for humorous headlines. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 133–142.

Nabil Hossain, John Krumm, Michael Gamon, and Henry Kautz. 2020. Semeval-2020 task 7: Assessing humor in edited news headlines. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 746–758.

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

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. In *International Conference on Learning Representations*.

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*.

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*.

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

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.