hub at SemEval-2021 Task 7: Fusion of ALBERT and Word Frequency Information Detecting and Rating Humor and Offense

Bo Huang
School of Information Science
and Engineering Yunnan University,
Yunnan, P.R. China
hublucashb@gmail.com

Yang Bai
School of Information Science
and Engineering Yunnan University,
Yunnan, P.R. China
baiyang.top@gmail.com

Abstract

This paper introduces the system description of the hub team, which explains the related work and experimental results of our team's participation in SemEval 2021 Task 7: Ha-Hackathon: Detecting and Rating Humor and Offense. We successfully submitted the test set prediction results of the two subtasks in the task. The goal of the task is to perform humor detection, grade evaluation, and offensive evaluation on each English text data in the data set. Tasks can be divided into two types of subtasks. One is a text classification task, and the other is a text regression task. What we need to do is to use our method to detect the humor and offensive information of the sentence as accurately as possible. The methods used in the results submitted by our team are mainly composed of ALBERT, CNN, and Tf-Idf algorithms. The result evaluation indicators submitted by the classification task are F1 score and Accuracy. The result evaluation index of the regression task submission is the RMSE. The final scores of the prediction results of the two subtask test sets submitted by our team are task1a 0.921 (F1), task1a 0.9364 (Accuracy), task1b 0.6288 (RMSE), task1c 0.5333 (F1), task1c 0.0.5591 (Accuracy), and task2 0.5027 (RMSE) respectively.

1 Introduction and Background

Perceiving humor has always been a unique ability of human beings. So what is the use of humor? The research results of Martin and Kuiper on humor show us the influence of humor on a person's physical and mental health (Martin, 2004; Kuiper et al., 2004). In recent years, the use of automated methods to detect humorous information in text has attracted widespread attention (Barbieri and Saggion, 2014; Reyes et al., 2012).

SemEval 2021 Task 7: Ha-Hackathon: Detecting and Rating Humor and Offense's task goal is to use automated techniques and methods to automatically detect humor and grade in the text. Besides, this task also needs to evaluate the offensive level of the text data. The task is divided into two parts, one part is the detection and evaluation related to humor. There are three subtasks in this part of the task. It involves text classification and regression. The other part is to assess the offensive level of the text data. This is a separate text regression task. The purpose is to predict how offensive the text will be to ordinary users. This task is an interesting challenge for the machine. Humor is a very subjective emotion. People with different cultural backgrounds and life experiences have different feelings about the same sentence. This task is to detect and score humor on the English data set. There are similar tasks and studies in other languages, such as humor scores on Spanish data from tweets (Castro et al., 2018; Chiruzzo et al., 2019).

In text detection and classification tasks, semi-supervised and supervised methods are widely used. Davidov et al. use a semi-supervised method to detect text from social media. The purpose is to detect whether the text data contains ironic information (Davidov et al., 2010). But these methods alone are not enough to make humor ratings on text data. We need to combine semantic information in context. The ELMo (Peters et al., 2018) method based on LSTM (Olah, 2015) overcomes the difficulty that the model cannot learn the context. In the follow-up work, an improved method of ELMo feature extractor appeared. The BERT (Devlin et al., 2018) model based on Transformer Encoder (Vaswani et al., 2017) achieved the best results in many NLP tasks.

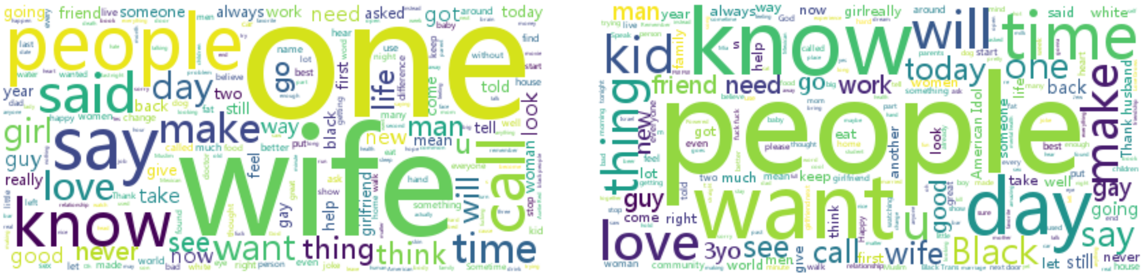
2 Data and Methods

In this section, we will introduce the data we use in the task and the models and methods we use.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1141–1145
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

ID	Text	Is humor	Humor rating	Humor controversy	Offense rating
35	Learn from the scars of others	0	0	0	0.05
119	What do you call a sad terrorist? A crisis	1	2.16	1	0.85

Table 1: The training set sample data we used in the task.



(a) The training data

(b) The test data

Figure 1: The word cloud diagram of the training and test data provided by the task organizer team. The result shown in the figure is the data after removing the stop words.

2.1 Data Description

The task organizer team provides each team with training data sets, validation data sets and test data sets related to the "Detecting and Rating Humor and Offense" task (Meaney et al., 2021). We analyze the structure and characteristics of the data sets. The training data set includes ID, Text, Is Humor, Humor Rating, Humor Controversy, Offense Rating. Among them, Is Humor, Humor Rating, Humor Controversy 3 tags are the three subtasks a, b, and c of task 1. Is Humor and Humor Controversy are two binary classification labels, consisting of 0 or 1. Humor Rating is a continuous value between 0-5. Offense Rating is the label of task 2. It is a continuous value between 0-5. The sentence length in the Text is different. Compared with the training data set, the test set only contains the above ID and Text parts. During the development phase, the task organizer also provides a test set. But we did not use this test set in our system, so we do not analyze the test set. We need to use our method to predict the values of Is Humor, Humor Rating, Humor Controversy and Offense Rating labels in the test set. Table 1 shows a sample of the data we used in the task.

8000 and 1000 different sample data constitute the training set and the validation set. The numbers of labels belonging to 1 and 0 in the training set Is Humor label are 4932 and 3068, respectively. The numbers of labels belonging to 1 and 0 in the training set Humor Controversy label are 2465 and 5535, respectively. The test set consists of 1000 different sample data. We use word cloud graphs to visualize text data. Text data in the training set and test set. The word cloud image clearly shows us the characteristics of word frequency distribution in the text data set. The figure shows the text data after the stop words are deleted. Figure 1 shows word frequency information in the training set and the test set.

2.2 Methods

Combined with the analysis and understanding of task description and task data set, we chose to develop an artificial neural network system based on the combination of ALBERT, Tf-Idf and CNN. We also tried to use the combination of BERT and Tf-Idf to verify its impact on the verification set. Both BERT and ALBERT (Lan et al., 2019) are pre-trained language models that are implemented based on the ideas and structure of Transformer. Compared with BERT, ALBERT not only has fewer parameters but also has the characteristics of parameter sharing between different layers. The pre-trained language model also occupies less memory space. Therefore, ALBERT is better than BERT in training effect. The CNN block we use in the system is mainly composed of two-dimensional

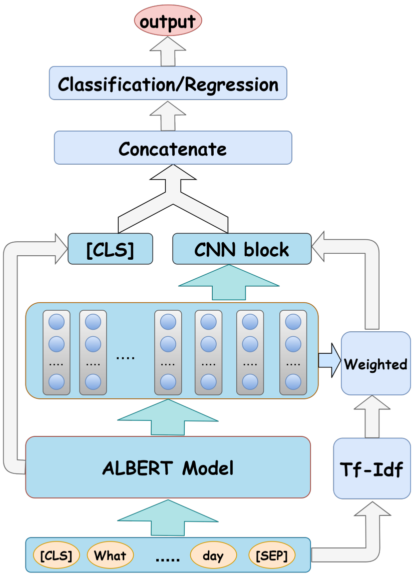


Figure 2: The model structure and data flow we used in the task.

convolution and two-dimensional maximum pooling. The convolution kernel has three (3, 4, 5) different sizes. The processed results of three convolution kernels of different sizes are connected as the output result of the CNN block.

In the system we use to predict the results of the test set. The first step is to input the preprocessed text data into the ALBERT model. At the same time, the text is processed with Tf-Idf to get Tf-Idf_output. In the ALBERT model, we will get two output values. One is [CLS] (the shape is [batch_size, hidden_size]) that contains the entire sentence information. The other is the last layer output of the ALBERT model last_layer_output (the shape is [batch_size, seq_length, hidden_size]). In the second step, we use Tf-Idf_output to perform a weighted operation on last_layer_output to get weighted_output (the shape is [batch_size, seq_length, hidden_size]). In the third step, we use weighted_output and last_layer_output respectively as the input of the CNN block. The two output results have the same shape as [CLS]. In the fourth step, we stitch together the results of the two CNN blocks obtained in the previous step and the results of [CLS] to obtain Concatenate_output (the shape is [batch_size, hidden_size*3]). In the fifth step, we use Concatenate_output as the input value of the classifier for the classification or regression task to obtain the predicted output result. Figure 2 shows our model structure and data flow.

3 Experiment and Results

In this section, we will introduce the data preprocessing methods and experimental settings we used in the task and the final results.

3.1 Data Preprocessing

Combining our understanding of tasks and data, we removed the stop words in the text data. For the stop word list, we use the stop word package provided by NLTK. Besides, to use the Tf-Idf algorithm to obtain a weighted output, and to ensure that the shape of the text code processed by the Tf-Idf algorithm is consistent with the shape of the ALBERT output, we removed the text code that exceeded the maximum sentence length. For those texts that are less than the maximum sentence length. For text encoding, we perform zero padding.

The validation set we get is unlabeled data, so the validation set we use in the training phase comes from part of the data in the training set. Randomly extract 20% from the training set as the validation set data we will use next.

3.2 Experiment setting

To test the influence of different systems on the prediction results of the task data set. We design several different models and observe the result scores of different models on the validation set. We adjust the parameters as much as possible to obtain the best results for each different model, so different models use different parameter combination settings.

The difference between BERT+Tf-Idf+CNN and the system we introduced above is only to replace the ALBERT model. BERT+Tf-Idf and ALBERT+Tf-Idf directly splice the output results of BERT and ALBERT with the weighted result of Tf-Idf. The other parts are the same as we introduced in section 2.2. BERT and ALBERT directly use their [CLS] output results. The task of classification uses the BCEWithLogitsLoss loss function. The regression task uses the MSELoss loss function.

- ALBERT+Tf-Idf+CNN: The epoch, batch size, maximum sequence length, and learning rate for the model are 5, 32, 70, and 3e-5, respectively.
- BERT+Tf-Idf+CNN: The epoch, batch size, maximum sequence length, and learning rate

Method	task1a	task1	task1b	task1c	task1c	task2
	F1	Acc	RMSE	F1	Acc	RMSE
ALBERT+Tf-Idf+CNN	**0.945**	0.943	**0.605**	0.522	**0.574**	**0.492**
BERT+Tf-Idf+CNN	0.930	**0.944**	0.620	**0.532**	0.561	0.502
ALBERT+Tf-Idf	0.921	0.929	0.617	0.545	0.554	0.490
BERT+Tf-Idf	0.925	0.932	0.627	0.528	0.564	0.491
ALBERT	0.915	0.927	0.634	0.532	0.544	0.510
BERT	0.917	0.923	0.625	0.542	0.551	0.497

Table 2: We use different strategies to get the scores on the validation set.

Method	task1a	task1a	task1b	task1c	task1c	task2
	F1	Acc	RMSE	F1	Acc	RMSE
Top1	0.982	0.985	0.496	0.494	0.630	0.412
Top2	0.975	0.980	0.498	0.470	0.628	0.419
Top3	0.960	0.968	0.521	0.470	0.627	0.423
Our team	0.921	0.936	0.629	0.533	0.559	0.503

Table 3: Part of the results in the leaderboard announced by the task organizer team. Among them, the results of Top1-Top3 are the combination of the scores of the top three in each subtask, not the scores of a team. The total number of participating teams in each of the four subtasks is 58, 50, 36, 48.

for the model are 4, 32, 70, and 4e-5, respectively.

- ALBERT+Tf-Idf: The epoch, batch size, maximum sequence length, and learning rate for the model are 5, 32, 70, and 3e-5, respectively.

- BERT+Tf-Idf: The epoch, batch size, maximum sequence length, and learning rate for the model are 4, 32, 70, and 4e-5, respectively.

- ALBERT: The epoch, batch size, maximum sequence length, and learning rate for the model are 4, 32, 70, and 3e-5, respectively.

- BERT: The epoch, batch size, maximum sequence length, and learning rate for the model are 4, 32, 70, and 3e-5, respectively.

4 Results and Analysis

The evaluation indicators announced by the task organizer team in this task are divided into classification tasks and regression tasks. The classification task uses F1 scores and accuracy scores. The regression task uses root mean squared error (RMSE).

We compare the results obtained by several different methods proposed in the experimental part. The scores of different systems on the same validation set are shown in Table 2. We have the following conclusions:

- Conclusion 1: Introducing additional word frequency information as part of the input information of the model will improve the score of our validation set.

- Conclusion 2: The score difference between ALBERT and BERT on our validation set is not large. But in the same parameters and data set, the training time of ALBERT is shorter than BERT.

- Conclusion 3: Adding a CNN block improves the result score. And the result scores of the two methods based on ALBERT and BERT have their advantages.

The ranking of the test set prediction results announced by the task organizer team uses accuracy scores and RMSE scores respectively. Classification tasks are ranked according to accuracy scores. Regression tasks are ranked according to RMSE scores. We finally submitted the test set prediction result score from the ALBERT+Tf-Idf+CNN system. Because its combined result in training time and score is better than the BERT+Tf-Idf+CNN system. The prediction result scores of the test set we submitted can be seen in Table 3. The scores of our system on the test set are lower than the scores of the top three. But the results of the test set prove that our system and method are feasible.

5 Conclusion

In this task related to humor detection and offensive evaluation, we propose an artificial neural network system that combines a pre-trained language model with Tf-Idf and CNN. Although our system is only in the middle of the ranking, we still successfully use our system to predict humor and offensive scores. We studied the contributions of 6 different systems and found that the combination of Tf-Idf and CNN improved the prediction scores of BERT and ALBERT. At the same time, the ALBERT-based system is superior to the BERT-based system in terms of time efficiency. In future work, based on the model we use in the task, we can try to fuse other types of word embedding information, and replace the CNN block with an LSTM block or other artificial neural networks.

References

Francesco Barbieri and Horacio Saggion. 2014. Automatic detection of irony and humour in twitter. In *ICCC*, pages 155–162.

Santiago Castro, Luis Chiruzzo, and Aiala Rosá. 2018. Overview of the haha task: Humor analysis based on human annotation at ibereval 2018. In *IberEval@ SEPLN*, pages 187–194.

Luis Chiruzzo, Santiago Castro, Mathias Etcheverry, Diego Garat, Juan José Prada, and Aiala Rosá. 2019. Overview of haha at iberlef 2019: Humor analysis based on human annotation. In *IberLEF@ SEPLN*, pages 132–144.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcasm in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116.

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*.

Nicholas A Kuiper, Melissa Grimshaw, Catherine Leite, and Gillian Kirsh. 2004. Humor is not always the best medicine: Specific components of sense of humor and psychological well-being. *Humor: International Journal of Humor Research*, 17.

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. *arXiv preprint arXiv:1909.11942*.

Rod A Martin. 2004. Sense of humor and physical health: Theoretical issues, recent findings, and future directions. *Humor*, 17(1-2):1–19.

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*.

Christopher Olah. 2015. Understanding LSTM networks.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12.

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