WUY at SemEval-2020 Task 7: Combining BERT and Naive Bayes-SVM for Humor Assessment in Edited News Headlines

Cheng Zhang
Graduate School of Fundamental Science and Engineering, Waseda University,
3-4-1 Okubo, Shinjuku-ku,
Tokyo, 169-8555 Japan

Hayato Yamana
Faculty of Science and Engineering,
Waseda University,
3-4-1 Okubo, Shinjuku-ku,
Tokyo, 169-8555 Japan

{zchelllo, yamana}@yama.info.waseda.ac.jp

Abstract

This paper describes our participation in SemEval 2020 Task 7 on assessment of humor in edited news headlines, which includes two subtasks, estimating the humor of micro-editd news headlines (subtask A) and predicting the more humorous of the two edited headlines (subtask B). To address these tasks, we propose two systems. The first system adopts a regression-based fine-tuned single-sequence bidirectional encoder representations from transformers (BERT) model with easy data augmentation (EDA), called "BERT+EDA". The second system adopts a hybrid of a regression-based fine-tuned sequence-pair BERT model and a combined Naive Bayes and support vector machine (SVM) model estimated on term frequency–inverse document frequency (TFIDF) features, called "BERT+NB-SVM". In this case, no additional training datasets were used, and the BERT+NB-SVM model outperformed BERT+EDA. The official root-mean-square deviation (RMSE) score for subtask A is 0.57369 and ranks 31st out of 48, whereas the best RMSE of BERT+NB-SVM is 0.52429, ranking 7th. For subtask B, we simply use a sequence-pair BERT model, the official accuracy of which is 0.53196 and ranks 25th out of 32.

1 Introduction

Humor, a high-level form of human communication, is omnipresent, including in the social media as well as real-life situations. The proper use of humor can have a positive impact on our lives. An automatic identification of humor can help in better understanding the structure and theory of humor. Moreover, correctly understanding humor is important for improving the performance of many natural language processing applications such as a sentiment analysis and intention mining.

To evaluate the progress of automatic humor assessment, SemEval-2020 task 7[1] aims to study how machines can understand humor generated by applying short edits to news headlines (Hossain et al., 2020). A dataset, called **Humicroedit** (Hossain et al., 2019), with a total of 15,095 English edited news headlines collected from Reddit (reddit.com) along with mean humor scores, was developed. With this dataset, humor is generated after a short editing, e.g., *"President vows to cut ~~taxes~~ hair"*. Subtask A focuses on the prediction of the mean humor of an edited headline, and subtask B involves predicting the funnier of two edited headlines.

To address this problem, we propose two systems[2]. The first (BERT+EDA) is a regression fine-tuned single sequence bidirectional encoder representations from transformers (BERT) model (Devlin et al., 2019) with easy data augmentation (EDA) tool[3], which is an implementation of the data augmentation theory of Wei and Zou (2019). The second (BERT+NB-SVM) is a hybrid of the regression fine-tuned sequence-pair BERT and a Naive Bayes-Support Vector Machine (NB-SVM) model (Wang and Manning, 2012) estimated on TFIDF features. Here we combine the strength of deep learning and classical machine learning. BERT+NB-SVM outperforms BERT+EDA. Both systems are described in the later sections,

[1] https://competitions.codalab.org/competitions/20970

[2] Source code for our model is published on https://github.com/HeroadZ/SemEval2020-task7

[3] https://github.com/jasonwei20/eda_nlp

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1071–1076
Barcelona, Spain (Online), December 12, 2020.

although the official result for subtask A is only from BERT+EDA. For subtask B, we simply use a sequence-pair BERT model. The official accuracy is 0.53196, which rankes 25th out of 32.

In the rest of this paper, studies related to automatic humor recognition are introduced in section 2. An overview of the system is presented in section 3. We then describe the experiments and results in section 4. Finally, we provide some concluding remarks in section 5.

2 Related Work

The task of automatic humor recognition refers to deciding whether a given sentence expresses a certain degree of humor. However, this remains a challenge (Attardo, 1994) because there is no universal definition of humor, and an understanding of a same sentence depends on personal background of the readers. Previous studies on humor recognition have focused on the binary classification problem (humorous or not). In Taylor (2004), a joke recognizer was developed to determine whether a discovered wordplay makes the text funny by learning the statistical patterns of text in N-grams. In Yang et al. (2015), semantic features designed to recognize humor based on four structures are described: incongruity, ambiguity, interpersonal effect and phonetic style. Moreover, the authors proposed an effective maximal decrement method to extract humor anchors in the text. In addition to linguistic features, some other studies have tried to make use of spoken or multimodal signals for improvement. For instance, in Purandare and Litman (2006), a classical supervised decision tree classifier with a set of acoustic-prosodic features and linguistic features is used to recognize humor in conversations from a comedy television show.

Many recent studies have attempted to utilize a neural network for humor detection. In De Oliveira and Rodrigo (2015), recurrent neural network (RNNs) and convolutional neural networks (CNNs) are applied to humor detection in a dataset on Yelp reviews. They found that the CNN model outperformed RNN with two points. In Chen and Soo (2018), a CNN architecture is constructed using highway networks and applied to a balanced large-scale dataset called "Pun of the Day", of both English and Chinese texts.

In this paper, we constructed systems based mainly on BERT and NB-SVM models. BERT is pretrained on unlabeled text (Wikipedia) through joint conditioning on both the left and right contexts in all layers. We chose BERT becacuse it has obtained state-of-the-art results on 11 natural language processing tasks, including text classification and regression datasets like GLUE (Wang et al., 2018). In addition we chose NB-SVM model based on the study in Wang and Manning (2012), which demonstrated that NB achieves better results for short snippets of sentiment tasks, whereas SVM is better for longer documents. An SVM model using NB log-count ratios as features performs well across different tasks. BERT focuses on the semantics, and NB-SVM focuses on linguistics, and this combination performed well at Human Annotation Challenge at IberLEF 2019 (Ismailov, 2019).

3 System Overview

3.1 BERT + EDA

The structure of BERT + EDA is shown in Figure 1. It is a regression fine-tuned single-sequence BERT with data augmentation (Wei and Zou, 2019) in simple words. Initially, we use an easy data augmentation (EDA) tool to create more data for training. EDA performs four operations on a given sentence, i.e., synonym replacement, random insertion, random swap, and random deletion. During the second step, we fine tune the BERT with the augmentation dataset. Here because we apply a single-sequence BERT, we replace the original word with the edit word. For example, if the original sentence is *"President vows to cut ⟨taxes⟩"* and the edit word is *"hair"*, the input should be *"President vows to cut hair"*. If the model is for sequence-pair training, the input should be a tuple containing *"President vows to cut taxes"* and *"President vows to cut hair"*. We then tokenize the test sentence and predict the result using the fine-tuned BERT model. Finally, we process the prediction to make it closer to the possible grade. For instance, we transform 1.333 to 1.4 and 1.233 to 1.2. As the reason for this processing, the mean grade should be within a limited range.

3.2 BERT+NB-SVM

The structure of BERT+NB-SVM is shown in Figure 2. As can be seen, the process on the left is for training and the process on the right is for predicting. In the training step, we fine-tune BERT model with the training dataset. The SVM model is then trained with NB log count ratios from TFIDF matrix of the training dataset. During the prediction step, the final score is the weighted sum of the fitted NB-SVM model and best fine-tuned BERT model. The best fine-tuned BERT model indicates the model that performs best among all models trained with different epochs and with different batch sizes. The weights of the best fine-tuned BERT model and NB-SVM model are 0.91 and 0.09, as derived from the grid search. The same processing made in BERT+EDA is also necessary during the final step.

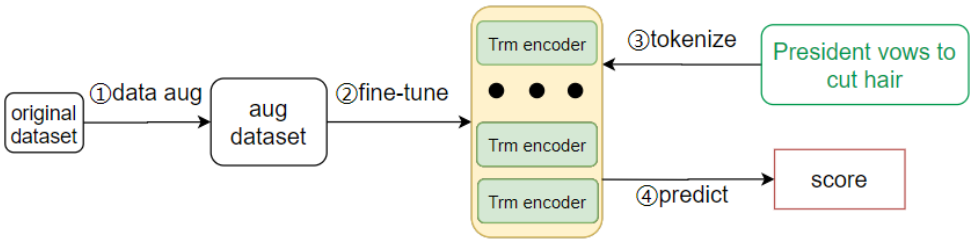

Figure 1: The structure of BERT+EDA

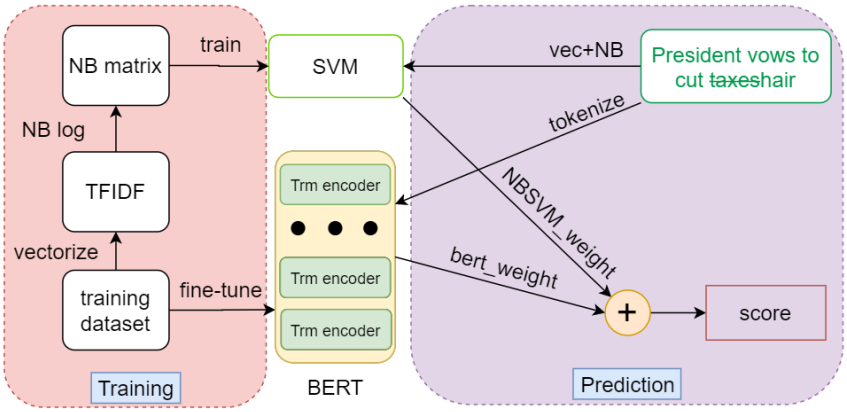

Figure 2: The structure of BERT+NB-SVM

4 Experiments and Results

4.1 Dataset and Experimental Settings

The statistics of the dataset[4] used in this task are shown in Table 1. The headlines used in subtask B are all from subtask A. Here we combine "train" and "dev" datasets for training. In subtask A, the performance of systems is evaluated based on root-mean-square deviation(RMSE). In subtask B, the accuracy is used as the metric.

Task	A			B		
Dataset	train	dev	test	train	dev	test
Number	9652	2419	3024	9381	2355	2960

Table 1: The statistics of the dataset.

[4]https://www.cs.rochester.edu/u/nhossain/humicroedit.html

In this work, we apply a pretrained BERT model in a Pytorch implementation using HuggingFace[5]. We use only the 12-layer BERT-base-cased model pretrained on cased English text. A large BERT model is not used because the dataset is not large. For the BERT tokenizer, we set the maximum length of single-sequence BERT to 96 and sequence-pair BERT to 160 because the longest sentence in the dataset has 74 tokens after tokenization. For the hyperparameters of SVM model, C is 1.0 and epsilon is 0.3, which are the default settings of SVR model. We trained all models except NB-SVM model with different epochs and batch sizes. The epoch ranges from 1 to 4, and the batch size is set to {32, 16, 8, 4}. For the EDA tool, the number of augmentation for each sentence is 4 and the change in percentage (alpha) is 0.05 by default. The random seed is 66. In the next subsection, we introduce the performance evaluation of our systems for each task, and the results of which are all reproducible from our code. Note that the maximum length for the BERT tokenizer of the official results are 300, whereas in the post-evaluation stage we decrease it to 96 and 160 for less memory usage and an improved performance. Because the parameters used in the official results differ slightly from the above parameters, and thus the following results are also different.

4.2 Performance Evaluation for Subtask A

The RMSE scores of the sequence-pair BERT model are shown in Table 2. Here, E and BS represent the epoch and batch size respectively. As shown in this table, the RMSE score of best model is 0.52642 which is trained in $e = 1$ and $bs = 4$. With an increase in the number of epochs, the RMSE score does not decrease, whereas with a decrease in the batch size, the RMSE score decreases. Therefore, we can observe that the batch size has a great impact on the results. Moreover, the data appear to be insufficient because overfitting occurs after two epochs.

E\BS	**bs=32**	**bs=16**	**bs=8**	**bs=4**
e=1	0.56554	0.53797	0.53862	**0.52642**
e=2	0.54039	0.53015	0.54117	0.53173
e=3	0.54451	0.55280	0.54662	0.55499
e=4	0.58413	0.56029	0.56518	0.56136

Table 2: The RMSE scores of sequence-pair BERT for subtask A.

E\BS	**bs=32**	**bs=16**	**bs=8**	**bs=4**
e=1	0.54199	0.54312	0.53396	0.53438
e=2	**0.53324**	0.54381	0.53830	0.54047
e=3	0.54728	0.54737	0.56014	0.55393
e=4	0.55964	0.55747	0.56742	0.56569

Table 3: The RMSE scores of single sequence BERT for subtask A.

The RMSE scores of single-sequence BERT are shown in Table 3. The RMSE score of the best model, which is trained with $e = 2$ and $bs = 32$, is 0.53324. It performs worse than sequence-pair BERT. The reason for which might be because the sequence-pair BERT learned the information of contrast in context. For example, we input both *"hair"* and *"taxes"* into the sequence-pair BERT, whereas for single-sequence BERT we only input *"hair"*. During the annotation stage, we assumed the edited headlines creating a large contrast obtains higher humor score. In addition, the overfitting still occurs during the training step.

Therefore, we introduced EDA to solve the overfitting problem. The combination of single-sequence BERT and EDA called "BERT+EDA". The EDA generates four short modified versions for each sentence. The RMSE scores of BERT+EDA are shown in Table 4. The RMSE score of the best model, which is trained in $e = 1$ and $bs = 16$, is 0.56248. The results show that the EDA is not completely helpful for datasets with more than 10,000 samples. Unfortunately, we submitted the output of BERT+EDA as last submission since we thought the evaluation will take the best result rather than the last submission.

The RMSE score of NB-SVM model is 0.56439. Because the best sequence-pair BERT model described above is trained using $e = 1$ and $bs = 4$, we combined it with the NB-SVM model (BERT+NB-SVM). The RMSE score of the BERT+NB-SVM is 0.52429 based on the weights described in Section 3.2. The results of all systems are presented in Table 5. All systems outperform the baseline. The best score is 0.52429 from BERT+NB-SVM, which combines the strength of BERT and NB-SVM.

[5]https://huggingface.co/transformers/pretrained_models.html

E\BS	bs=32	bs=16	bs=8	bs=4
e=1	0.57417	**0.56248**	0.57541	0.56803
e=2	0.57102	0.57212	0.57898	0.56313
e=3	0.57138	0.57297	0.57026	0.57035
e=4	0.58044	0.56966	0.56339	0.56325

Table 4: RMSE scores of BERT+EDA for subtask A.

System	RMSE score
baseline	0.57471
NB-SVM	0.56439
single BERT + aug	0.56248
single BERT	0.53324
pair BERT	0.52642
BERT + NBSVM	**0.52429**

Table 5: RMSE scores of all systems for subtask A.

4.3 Performance Evaluation for Subtask B

In subtask B, we adopt two strategies to handle the classification problem. The first is utilizing the systems used in subtask A to predict two humor scores and compare them. The second is formulating a sequence-pair BERT classifier.

The inputs of the sequence-pair BERT classifier are the edited headlines without original words. For example, one is *"#WomensMarch against Donald Trump around the ~~world~~kitchen"* and the other *"#WomensMarch against ~~Donald Trump~~men around the world"*. The accuracy scores of the sequence-pair BERT classifier (second strategy) are shown in Table 6. The accuracy of the best model, which is trained using $e = 4$ and $bs = 8$, is 0.53311.

E\BS	bs=32	bs=16	bs=8	bs=4
e=1	0.43108	0.45338	0.45473	0.48750
e=2	0.45203	0.45169	0.52466	0.43986
e=3	0.44595	0.47534	0.46047	0.51655
e=4	0.46250	0.51385	**0.53311**	0.48480

Table 6: Accuracy scores of sequence-pair BERT classifier for subtask B.

System	Accuracy
baseline	0.43547
BERT + NBSVM	0.46926
NB-SVM	0.48209
single BERT + aug	0.49696
single regress BERT	0.51149
pair regress BERT	0.53007
pair BERT clf	**0.53311**

Table 7: Accuracy scores of all systems for subtask B.

An overview of the results for subtask B is shown in Table 7. The systems that perform well on subtask A do not achieve a high accuracy. BERT+NB-SVM achieved the lowest score of 0.46926, which is lower than the NB-SVM model (0.48209). Although the accuracy of both regression BERT models is above 0.5, despite the best accuracy from sequence-pair BERT classifier (0.53311) outperforming the baseline by 10%, the model still obtained a low rank among the other submissions. We analyzed the accuracy of this model for each class, i.e., 0% for class 0, 61% for class 1 and 59% for class 2. This shows that our systems struggle with recognizing the same humorous sentences.

5 Conclusion

In this study, we developed and compared systems constructed using BERT and NB-SVM models to deal with the humor assessment in newly edited headlines at SemEval 2020 Task 7. The combination of the sequence-pair regression BERT model and the NB-SVM model performed well for subtask A. However, the EDA tool was not completely successful in improving the outcome. Moreover, our solutions struggle with the comparison of two sentences with same humor score. The official RMSE for subtask A is 0.57369 and ranked 31st out of 48 whereas the best RMSE of our submission is 0.52429, which ranks 7th. The official accuracy for subtask B is 0.53196, which ranks 25t out of 32. In future studies, we aim to focus on the using of a large BERT model and the development of other useful data augmentation tools.

Acknowledgements

The authors would like to thank the reviewers for their insightful comments and constructive suggestions on improving this work.

References

S. Attardo. 1994. *Linguistic Theories of Humor*. Humor research. Mouton de Gruyter.

Peng-Yu Chen and Von-Wun Soo. 2018. Humor recognition using deep learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 113–117, New Orleans, Louisiana, June. Association for Computational Linguistics.

Luke De Oliveira and Alfredo L Rodrigo. 2015. Humor detection in yelp reviews. *Retrieved on December*, 15:2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

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, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Nabil Hossain, John Krumm, Michael Gamon, and Henry Kautz. 2020. Semeval-2020 Task 7: Assessing humor in edited news headlines. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain.

Adilzhan Ismailov. 2019. Humor analysis based on human annotation challenge at iberlef 2019: First-place solution. In *IberLEF@SEPLN*.

Amruta Purandare and Diane Litman. 2006. Humor: Prosody analysis and automatic recognition for f*r*i*e*n*d*s*. In *In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06*, pages 208–215, 01.

Mazlack Taylor. 2004. Computationally recognizing wordplay in jokes. *Cognitive Science - COGSCI*, 01.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea, July. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, Hong Kong, China, November. Association for Computational Linguistics.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2367–2376, Lisbon, Portugal, September. Association for Computational Linguistics.