Humor Detection based on Paragraph Decomposition and BERT Fine-Tuning

Hao Yang,[†*] **Yao Deng,**[†] **Minghan Wang,**[†] **Ying Qin**[†] **Shiliang Sun,**[‡]
[†]Huawei Translate Service Center, Beijing, China
[‡]East China Normal University, Shanghai, China
{yanghao30, dengyao3, wangminghan, qinying}@huawei.com, slsun@cs.ecnu.edu.cn

Abstract

In this paper, we propose an approach for humor recognition that is usually formalized as text classification or regression tasks. Algorithms are developed to distinguish whether a text sequence is humorous, or how funny a joke is. Our approach is mainly developed on pre-trained BERT (Devlin et al. 2019), fine-tuned on the task specific dataset. We further propose a heuristic data augmentation method by decomposing a long paragraph into a pair of consecutive paragraphs for three purposes. 1) Increasing the size of the training set. 2) Improving the performance of long distance context modelling of BERT. 3) Automatically find the punchline of a joke if exists. Our approach is evaluated on three datasets including CCL2019 Chinese Humor Computation (XIAONIU Cup) dataset in Chinese, FUN (Blinov, Bolotova-Baranova, and Braslavski 2019) dataset in Russian and HAHA (Chiruzzo et al. 2019) dataset in Spanish, which achieves competitive results on all of them. The experimental results demonstrate that our approach could effectively detect jokes and tag the punchline if exists, in different languages.

Introduction

Machine learning has been adopted in computational linguistic for understanding natural languages for several decades, with the development of representation learning, rich semantics can be encoded into the dense vectors named as embedding, which significantly improves the ability of algorithms in understanding fine-grained emotions, for example, detecting humor from sentences. There can be many industrial applications of humor detection such as language understanding in dialogue system and sentiment classification in social network platforms. At the same time, for any question answering systems, understanding the intent of the user requires fine-grained analysis, for example, detecting user's moods with humor detection.

Humor detection tasks can often be divided into two stages. The first one can be considered as a binary classification task aims to classify whether a sentence is a joke (e.g. HAHA) or whether the joke is written by a human or an algorithm (e.g. XIAONIU). The second task is for multi-class classification or regression aims to assess the funniness level of a joke (e.g. XIAONIU and HAHA).

There can be many algorithms to solve these problems such as conventional machine learning algorithms like TF-IDF representation with SVM or deep learning based like paragraph encoder + dense classifier. However, most of these algorithms are typically designed for universal tasks without considering about the difference (e.g. the paragraph structure) between humor detection and other document classification tasks.

From a linguistic perspective, there are two critical features that are often found in jokes, which inspire us to model them in our algorithm and provide specific optimization for the task:

- Good setup and a punchline is the core of many jokes. The setup can be considered as the background of a story, and the punchline is the surprise or the exception that is commonly contradict to intuition, which is the trigger to make the reader laugh. The punchline often appears at the ends of the joke and should be short enough. Therefore, we may try to decompose the joke and model the setup and the punchline separately.
- The topic of the joke determines whether it is funny for most of the people. Social events, politics and daily life are mostly used as materials to write a joke, which means there are usually commonsense in the joke and requires prior knowledge to understand the conflict in the punchline. Therefore, a pre-trained language model is fairly appropriate for this task as it could provide better language representation learned from large corpus.

Therefore, we propose a method for humor detection which can be described as three stages. 1) Data augmentation with paragraph decomposition. 2) Fine-tuning BERT with task specific label. 3) Ensemble for the inference. The contribution of our work can be summarized as following:

- We propose a data augmentation method named paragraph decomposition which is specifically appropriate for humor detection tasks based on pre-trained BERT(Devlin et al. 2019).
- We perform experiments on the dataset of three languages, which achieves competitive results comparing

with the BERT baseline.

- We propose the single model ensemble which is able to make more reasonable prediction during inference.

Related Work

Recent studies on humor detection are diversity. Some researchers focuses on employing state-of-the-art studies like BERT (Devlin et al. 2019) to make better predictions, others attempts to improve lighter networks like LSTM (Hochreiter and Schmidhuber 1996) and CNN (Krizhevsky, Sutskever, and Hinton 2012) or even conventional machine learning algorithms to compete with deep neural network models. At the same time, researchers are publishing more high quality datasets in different languages which provides significant help in this area.

(Weller and Seppi 2019) propose a BERT based humor detection model, fine-tuned on corpus collected from Reddit, Short Jokes and Pun of the Day (Yang et al. 2015), which achieves significant improvement on the performance comparing with many CNN based models.

(Chiruzzo et al. 2019) summaries a series of works from teams who build models and conduct experiments on HAHA dataset in the IberLEF 2019. The first place (Ismailov 2019) propose the method based on a pre-trained multilingual BERT, and further pre-train it on the domain dataset. Finally, the model is fine-tuned with task specific labels. Apart from that, they combine the prediction of Naive Bayes with TF-IDF and NN outputs with logistic regression to produce the final prediction, which achieves the best result among all teams. Other teams also follows the framework by combining deep pre-trained models with conventional algorithms to acquire competitive predictions, which demonstrate the power of pre-trained language models.

(Blinov, Bolotova-Baranova, and Braslavski 2019; Chiruzzo et al. 2019; Yang et al. 2015) contributes large corpus in different languages like Russian and Spanish, which provides chances for researchers to build and evaluate their models on more diverse datasets. At the same time, they evaluate their datasets with proposed models and make detailed analysis which successfully demonstrates the good quality of the corpus.

By reviewing previous works and analyzing their results, we choose to follow a similar pipeline and start our work based on the pre-trained BERT. Furthermore, we decide to choose three types of datasets in different languages to investigate whether the phenomenon of punchline exists in jokes of different languages.

Approach

In this section, we introduce the detail of our method in three stages and discuss the advantage of our method comparing with others.

Paragraph Decomposition

We have briefly introduced the feature of a joke in the introduction section and pointed out the importance of the punchline. However, there is no publicly available large dataset with exact labeled location of the punchline sentence, which prevents us from decomposing the setup and the punchline directly. Therefore, for a joke, we permutes all terminal punctuations and insert the [SEP] token after them to decompose original paragraph into pairs of paragraphs.

More formally, assuming there are M_i terminal punctuations in a joke J_i, we can have M_i ways to cut the original joke into two pieces from the m-th punctuation, and resulting in $J_i^* = \{J_i^{(1)}, ..., J_i^{(M)}\}$ where the decomposed joke $J_i^{(m)}$ can be denoted as $J_i^{(m)} = [\text{[CLS]}, t_1, ..., t_m^{\text{PUNC}}, \text{[SEP]}, ..., t_n]$ with n tokens, and $|J_i^*| = M_i$. By doing this, we can enlarge the original data set to $\sum_i^N \sum_m^{M_i} J_i^{(m)}$ pairs of augmented jokes, and each augmented joke can be considered as a paragraph pair with same label as the original joke J_i. The idea of creating paragraph pairs is inspired by BERT but the application is different from BERT, in BERT, pairs of sentences may have 15% of probability to come from different source and the model have to predict whether they are consecutive, but in our work, two paragraphs are exactly come from same original joke and consecutive, [SEP] is only a placeholder for split.

Besides representing the punchline and enlarging the training set, the decomposition can also be used to optimize the performance for BERT while encoding long paragraph ($|J_i| > 300$). From the experiment, we find that treating a long sequence as a single paragraph (without [SEP] in the middle) will dramatically drop the performance of BERT in a humor classification task, however, by adding [SEP] at the appropriate location, the performance can be optimized. We assume that in the pre-training, the [SEP] could affect the self-attention to attend tokens in the pre-/post-sentence separately, which somewhat decreases the context length. Therefore, in our task, long jokes can be break with [SEP] to produce better representation, especially when [SEP] is located at the punchline.

Fine-Tuning with BERT

Same as other document classification tasks, we fine-tune BERT on the humor detection dataset and define task specific prediction heads, denoted as function f, the input of the prediction head is the concatenation representation of the first token ([CLS]) from the last four layers. The output dimension is dependent on specific tasks. We use weighted cross-entropy loss to deal with imbalanced datasets, the label weights are calculated as follows:

$$w_c = \frac{N}{N_c \times C} \tag{1}$$

where N is the number of samples in the training set, C is the number of classes (e.g. 2 for binary classification) and N_c is the number of samples classified as c.

More formally, we define the prediction as:

$$\hat{y} = f(x; \theta_f) \tag{2}$$

and define the loss as:

$$L(\hat{y}, y) = -\frac{1}{N} \sum_i^N \sum_c^C w_c y_{i,c} \log P(y_{i,c}; \theta_f, \theta_{\text{BERT}}) \tag{3}$$

Dataset	Train	Dev	Test
XIAONIU (Task 1)	16,420	1,026	4,106
XIAONIU (Task 2)	16,671	1,042	4,172
FUN	251,415	N/A	61,794
HAHA	24,000	N/A	6,000

Table 1: The detailed sample size of the datasets

Note that the parameters of BERT aren't frozen and can be jointly trained during the fine-tuning process.

Single Model Ensemble

Different from traditional ensemble strategies, we perform a data-level ensemble which works on the augmented data and only use a single model. As we mentioned in the paragraph decomposition section, a joke can be augmented to M versions and each of them will be predicted and produce a prediction $\hat{y_m}$, therefore, while predicting the label of the joke J, we perform max-pooling over the probability for each class among all possible decomposition, denoted as:

$$P(y_c) = \max_{m} P(y_c^m) \tag{4}$$

The reason of using max-pooling rather than average-pooling is that we believe there must be one decomposition that can be correctly split the setup and the punchline, which will be likely to produce a more confident result (i.e. higher probability score on class c), therefore, we choose to believe the most likely one rather than voting.

Experiments

In this section, we introduce the details of the datasets, the baseline methods, as well as the experimental setup.

Data

We perform experiments on three following datasets organized in three languages respectively. The detail can be found in Table 1

XIAONIU This dataset has two subsets where the first one is composed of 21,552 jokes either written by human or generated by algorithms, the task is to distinguish machine generated jokes from human written ones. 21,885 jokes in the second subsets are labeled in three levels with an approximate distribution of 2:3:1 and will be formulated as a tri-class classification problem. Note that the golden labels of development set and test set are not released, and can only be assessed by the competition organizer. The experimental results reported later is from the test set on the leaderboard[1]. For this dataset, we use the pre-trained BERT-base-chinese model as the encoder. However, we found that in the second subtask, there are about 70 jokes are overlength (more than 512 tokens), thus are removed from the training set. During validation and testing, overlength jokes are trimmed to 512 tokens. F1-score is used in both sub tasks as the evaluation metric.

Method	XIAONIU		FUN	HAHA
	Task1	Task2		
BERT (w/o PD)	0.8930	0.4889	0.9081	**0.7932**
BERT (w/ PD)	**0.8968**	**0.4936**	**0.9102**	0.7926

Table 2: Performance of different method evaluated on three dataset. Values are F1-scores and PD is the abbreviation of paragraph decomposition.

FUN FUN is proposed in (Blinov, Bolotova-Baranova, and Braslavski 2019), mainly collected from several Russian social network websites, it only contains binary labels (i.e. classifying whether a paragraph is humorous). Note that FUN is the largest dataset in our experiment, consisting of more than 313,210 samples, where 1877 are manually labeled and considered as golden truth. We use the train/test splits provided by the dataset in the experiment. BERT-base-multilingual-cased is used to encode the corpus, and F1-score is the evaluation metric.

HAHA HAHA (Chiruzzo et al. 2019) is a Spanish corpus collected from twitter for the competition of IberLEF 2019. There are 30,000 samples where 11,595 tweets are labeled as humorous (38.7%). The humorous tweets are further annotated with real number scores in the range of 1 to 5. We only do the first task (i.e. binary classification) aims to perform convenient comparison among three datasets with F1-score. We also use pre-trained BERT-base-multilingual-cased to encode the corpus.

Experimental Setup

The BERT model we used is implemented with transformers[2] developed by huggingface. We use PyTorch[3] to implement the classification/regression layer after the BERT encoder. The model is trained on 4 Titan Xp GPUs where each has 12 GB memory, the batch size is set to 96. We use the AdamW (Loshchilov and Hutter 2019) as the optimizer with the peak learning rate of 1e-4.

Analysis

The experimental results is shown in Table 2, where the BERT without decompose is our baseline method. We can see that there are improvements for the XIAONIU and FUN dataset, which demonstrates the effectiveness of paragraph decomposition. However, for the HAHA dataset, the performance of the decomposed version is slightly below the baseline, reasons will be analyzed in following paragraphs.

We pick a positive case from XIAONIU dataset showing that the model could make correct prediction when the joke is decomposed:

- 你的儿子，丈夫或姐夫会找到你丢失的针头......[SEP] 赤脚走路时。

Which can be translated as "Your son, husband or brother-in-law will find your lost needle...[SEP] when they walk barefoot.". The model trained with augmented data correctly pre-

[1] https://github.com/DUTIR-Emotion-Group/CCL2019-Chinese-Humor-Computation

[2] https://github.com/huggingface/transformers

[3] https://pytorch.org/

dicts that it is a joke when decomposed by the [SEP], nevertheless the baseline mode makes an incorrect prediction.

Another positive case is also from XIAONIU, which indicates that correctly decomposing the joke from the start of the punchline could produce higher probability of the correct class:

- 猫似乎只是在削尖他们的爪子。实际上，他们正在锻炼腿部肌肉。[SEP] : 0.55
- 猫似乎只是在削尖他们的爪子。[SEP] 实际上，他们正在锻炼腿部肌肉。 : 0.61

Which can be translated as " Cats seem to be just sharpening their claws. [SEP] In fact, they are exercising leg muscles.". Where the probability $P(y = \text{this is a joke}|x)$ is shown after the sentence. We can see that decomposing the joke from the start of the second sentence achieves higher probability and the second sentence is actually the punchline of this joke.

To explain the reason that our approach produces unsatisfactory results on HAHA, we also pick some cases showing that the tweets published in HAHA is relatively unclean, with noisy characters like hashtags or being barely readable even by human:

- Tu? Gustarme? JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA JA J Tengo que disimular un poco mas.
- #20CosasQueHacerAntesDeMorir: Ensearles la diferencia entre: -Hay de haber -Ah de lugar -Ay de exclamar -Ai se eu te pego.
- Rt con el pollo asado #PremiosFenix

Where repeatedly appeared "JA" and hashtags may corrupt the paragraph decomposition algorithm and produce unreasonable paragraph pairs. At the same time, BERT is not pretrained on tweets which means the token representations of HAHA is insufficient to encode correct semantics.

Conclusion

We propose an approach for detecting humors from a paragraph, which is built upon pre-trained BERT, fine-tuned on task specific data and improved by the paragraph decomposition and single model ensemble. Through the experimental results and case studies, we demonstrate that our approach could make substantial improvement on the performance comparing with BERT baseline. Our approach achieves competitive results on two datasets from different languages. However, we found that there is still room for improvement on uncleaned datasets like tweets, which will become our future work and might be improved by a further pre-training on in-domain corpus. In addition, better decomposition strategies can also be a choice for further investigation.

References

Blinov, V.; Bolotova-Baranova, V.; and Braslavski, P. 2019. Large dataset and language model fun-tuning for humor recognition. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 4027–4032.

Chiruzzo, L.; Castro, S.; Etcheverry, M.; Garat, D.; Prada, J. J.; and Rosá, A. 2019. Overview of HAHA at iberlef 2019: Humor analysis based on human annotation. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019.*, 132–144.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186.

Hochreiter, S., and Schmidhuber, J. 1996. LSTM can solve hard long time lag problems. In *Advances in Neural Information Processing Systems 9, NIPS, Denver, CO, USA*, 473–479.

Ismailov, A. 2019. Humor analysis based on human annotation challenge at iberlef 2019: First-place solution. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019.*, 160–164.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, 1106–1114.

Loshchilov, I., and Hutter, F. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Weller, O., and Seppi, K. D. 2019. Humor detection: A transformer gets the last laugh. *CoRR* abs/1909.00252.

Yang, D.; Lavie, A.; Dyer, C.; and Hovy, E. H. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 2367–2376.