XSYSIGMA at SemEval-2020 Task 7: Method for Predicting Headlines' Humor Based on Auxiliary Sentences with EI-Bert

Jian Ma , Shu-Yi Xie , Mei-Zhi Jin, Lian-Xin Jiang , Yang Mo , Jian-Ping Shen
Ping An Life Insurance
AI Department
ShenZhen, China
{MAJIAN446,XIESHUYI542,EX-JINMEIZHI001,JIANGLIANXIN769,
MOYANG853,SHENJIANPING324}@pingan.com.cn

Abstract

This paper describes xsysigma team's system for SemEval 2020 Task 7: Assessing the Funniness of Edited News Headlines. The target of this task is to assess the funniness changes of news headlines after minor editing and is divided into two subtasks: Subtask 1 is a regression task to detect the humor intensity of the sentence after editing; and Subtask 2 is a classification task to predict funnier of the two edited versions of an original headline. In this paper, we only report our implement of Subtask 2. We first construct sentence pairs with different features for Enhancement Inference Bert(EI-Bert)'s input. We then conduct data augmentation strategy and Pseudo-Label method. After that, we apply feature enhancement interaction on the encoding of each sentence for classification with EI-Bert. Finally, we apply weighted fusion algorithm to the logits results which obtained by different pre-trained models. We achieve 64.5% accuracy in subtask2 and rank the first and the fifth in dev and test dataset[1], respectively.

1 Introduction

Humor detection is a significant task in natural language processing. Most of the available humor datasets treat it as a binary classification task (Khodak et al., 2017; Davidov et al., 2010; Barbieri et al., 2014), i.e., examining whether a particular text is funny. It is interesting to study whether a short editing of a text can turn it from non-funny to funny. In this paper, we examine whether the headline is funniness or which word substitution is more funniness after a short editing. Such research helps us to focus on the humorous effect of word changes. Our goal is to determine how the machine understands the humor generated by such brief editing and to improve the accuracy of model prediction by different strategies, including rewriting the input sentences of model, enhancing local inference of features and applying the strategies of data augmentation and model ensemble.

The competition dataset comes from Humicroedit (Hossain et al., 2019), a novel dataset for research in computational humor. Each edited text headline is labeled into a score of 0-3 by 5 judges with an average final score. The competition consists of two subtasks: Subtask 1 is to predict the average funniness of the edited headlines given the original headlines and the edited headlines and Subtask 2 is to predict which edited text is more funniness in comparison given the original headlines and two edited ones. This competition is significant because it is helpful for the task of generating humorous texts (Hossain et al., 2017), which can be applied to chatbots and news headline generation.

In this paper, we only report our implementation on Subtask 2. We conduct several key data preprocessing, including data cleaning, constructing different sentence pair inputs, delivering semi-supervised clustering of the replacement words, and adding clustering labels as the features of the model inputs. After that, we apply Bert to obtain the word embeddings of the two input sentences and calculate the difference of the feature information with soft alignment. Next, we enhance the local inference information and concatenate them into a sequence to the Softmax layer for classification. Finally, we apply data augmentation and the Pseudo-Label iterative learning strategy to enrich the data information from the embeddings of three pre-trained models, Bert, XLNET, and RoBerta. Cross validation and weighted voting ensemble are conducted to improve the model performance.

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

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1077–1084
Barcelona, Spain (Online), December 12, 2020.

The contributions of this paper can be summarized as follows:

(1) We creatively construct sentence pair inputs with different sentence representations and generate additional feature representations from SLPA algorithm.

(2) We propose a new model framework EI-Bert which can improve the feature information interaction of the Bert's output.

2 Related Work

Recently, humor detection has become a hot research topic. Khodak et al. introduces the Self-Annotated Reddit Corpus (SARC), a large corpus for training and evaluating the task of sarcasm detection. Davidov et al.(2010) apply semi-supervised learning technique to identify sarcasms on two very different datasets and discuss the differences between the datasets and the algorithms. Barbieri et al.(2014) investigate the automatic method of detecting irony and humor in social networks. They cast the problem into a classification problem and propose a rich set of features and text representation to train the classifier. Kiddon et al.(2011) detect double-entendres, address this problem in a classification approach that includes features that model those two characteristics.

As a powerful semantic information feature extractor proposed by Devlin et al.(2018), Bert has two significant points that we can utilize: (1) Feature extraction with Transformer encoder, and training with MLM and NSP strategy pre-training; (2) The two-stage model of pre-training for large data scale pre-training and fine-tuning training for specific tasks. It demonstrates the effectiveness of bidirectional pre-training for language representation. Unlike the previously used unidirectional language models for pre-training, Bert applies a masked language model to achieve pre-trained deep bidirectional representations, which is the first representation model based on fine-tuning and achieves the most advanced performance on a large number of sentence-level and token-level tasks, also stronger than many task-oriented architectures.

Nowadays, researchers have proposed to construct auxiliary sentences in NLP tasks to enrich the input of Bert. For example, Sun et al.(2019) apply Bert to optimize fine-grained emotion classification and compare the experimental results of single sentence classification and sentence pair classification based on Bert fine-tuning, analyze the advantages of sentence pair classification, and verify the validity of conversion method. Keskar et al.(2019) propose to unify the QA problem and the text classification problem into a reading comprehension problem with the help of auxiliary data to enhance the model performance, and demonstrate that Bert is more suitable to handle Natural Language Inference (NLI) dataset. Clark et al.(2019) also show that the Bert model can learn more semantic information from inferential data. Qiao et al.(2019) concludes that if Bert is only adopted as a feature expression tool, the input side of Bert is just to enter Question or Passage separately, and take out the [CLS] mark of Bert high-level as the semantic representation of Question or Passage, this method is far less effective than entering Question and Passage on the Bert side at the same time which let Transformer do the matching process of Question and Passage.

3 System overview

3.1 Auxiliary sentences and data augmentation

In the system, we first conduct data cleaning, including word stem extraction, lexical reduction, spelling error correction, and punctuation processing. We construct auxiliary sentences by data augmentation through the replacement word rewriting and Pseudo-Label data generation via semi-supervised learning. Then, we form the auxiliary sentences as sentence pair as the input of Bert. As illustrated in Fig. 1, we construct three types of sentence pair inputs, where TextA is the first input sentence, TextB is the second input sentence.

As depicted in Fig. 1, given the original input sentence, "I am done fed up with California , some conservatives look to texas", shown in the lower left corner. The edited input sentence is "I am done fed up with California , some vagrants look to texas". The word "conservatives" is changed to "vagrants" which lies in TextA. Similarly, the word "California" is changed to "cakepan" which lies in TextB. Next, considering the high repetition of information between the original sentence and the edited sentence,

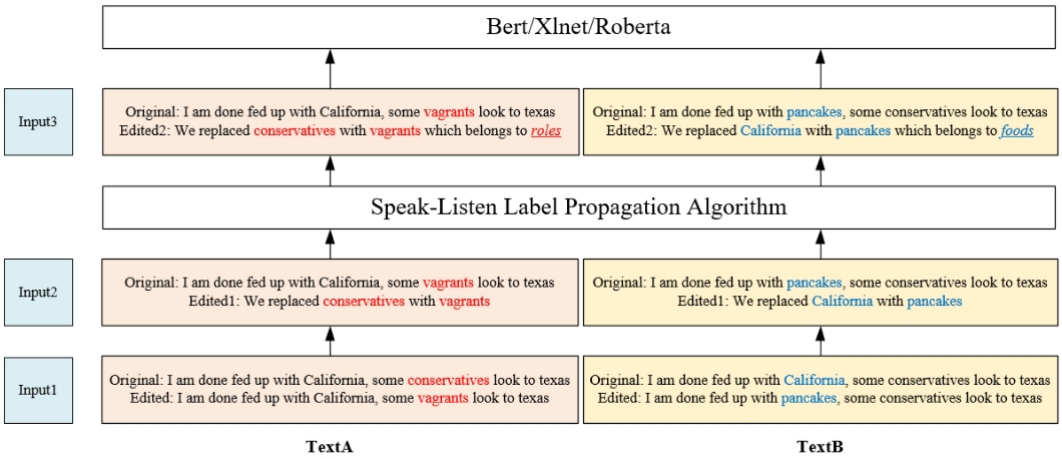

Figure 1: Three different sentence pair inputs

we construct new phrases, "We replaced conservatives with vagrants" and "We replaced California with pancakes", to make better use of the relationship between the sentences in Input2.

Finally, we group the replacement words into 50 clusters via the SLPA (Speaker-listener Label Propagation Algorithm), an extension of the LPA algorithm for community discovery (Yuchen et al., 2019). The input of the model is shown as input3, we add the clustering information of the replacement words.

Furthermore, we perform data augmentation. As shown in Fig. 2, TextA and TextB are the input sentences with minor editing. The score comes from the dataset which annotated from expert. We select the score of funniness greater than a certain threshold. We set the threshold to 0.8 to fit the dataset. Since the score of TextA and TextB are both greater than 0.8, then two other sentence pairs can be constructed. One is that TextA is unchanged, and the corresponding sentence adopt the original sentence which corresponds to Augmentation1 in the figure, since the score of TextA is greater than the score of TextB, the label value is 1. Similarly, TextB is unchanged, and the corresponding sentence adopt the original sentence which corresponds to Augmentation2 in the figure. The thresholds can be adopted slightly larger, which can make the sentence to semantic information gap larger.

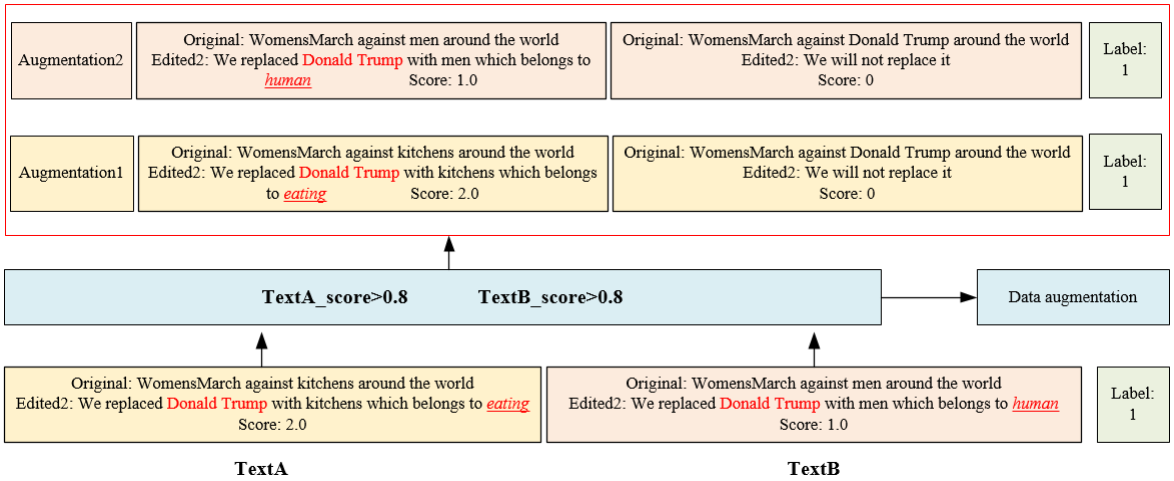

Figure 2: Data augmentation with certain threshold

As shown in Fig. 3, we consider improve our model's output by using an existing model to label non-labeled data and select the class with the largest predicted probability as Pseudo-Label. Then, we add the weak label data to the training process and add the weak loss to the original CE loss. The week loss is defined as follows:

$$L_w = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{m} L(y_i^j, f_i^j), \tag{1}$$

where n is the number of mini-batch in unlabeled data for SGD, m is the number of classes, y_i^j is the pseudo-label of that for unlabeled data, f_i^j is the output units of j's sample in unlabeled data. We iterate not only on the model, but also on the predicted Pseudo-Label, which is essentially a semi-supervised learning.

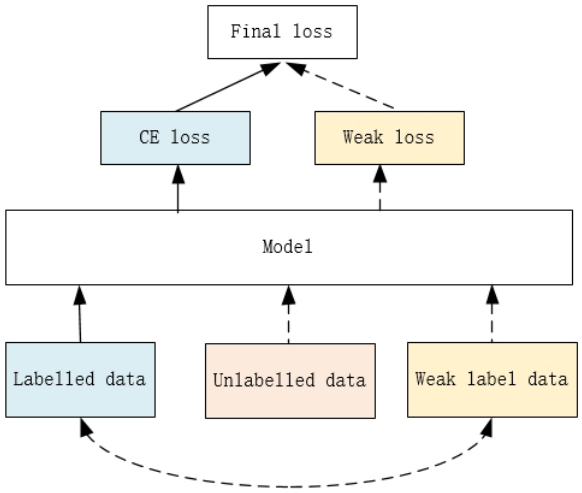

Figure 3: Pseudo-Label with labeled and unlabeled data simultaneously

3.2 Model Description

Natural Language Inference is mainly to determine the relationship between two sentences. In order to compare the funniness of the two sentences after editing, we naturally transform the Sentence Pair Classification task into a textual inference task to infer the relationship between the two input sentences. The model applies the Bert pre-trained model for encoding. It calculates the difference with sentence encoding information. We then concatenate this encoding with the original vectors to enhance the difference. Finally, we send the encoding vectors to the Max-Pooling and Softmax layers for classification. We name this procedure as Enhancement Inference Bert (EI-Bert), which is shown in Fig. 4.

We construct and augment the sentence pairs as in Fig. 1 and Fig. 2 into the Bert model and obtain the encoding information of the words in each sentence from the Bert's sequence output. We obtain the vector $\bar{a}_i$, $\bar{b}_j$ from Bert, where $\bar{a}_i$ is the input TextA's i_{th} token's representation vectors from Bert's last hidden state vectors, and $\bar{b}_j$ is the input TextB's j_{th} token's representation vectors from Bert's last hidden state vectors. We then calculate the similarity between the two sentence words. The attention weight is computed by

$$Att_{ij} = \bar{a}_i^T \bar{b}_j. \tag{2}$$

After that, we compute the weighted summation of $\tilde{a}_i$ and $\tilde{b}_i$ by

$$\tilde{a}_i = \sum_{j=1}^{l_b} \frac{exp(Att_{ij})}{\sum_{k=1}^{l_b} exp(e_{ik})} \bar{b}_j, i \in [1, ..., l_a], \tag{3}$$

$$\tilde{b}_i = \sum_{j=1}^{l_a} \frac{exp(Att_{ij})}{\sum_{k=1}^{l_a} exp(e_{ik})} \bar{a}_j, i \in [1, ..., l_b]. \tag{4}$$

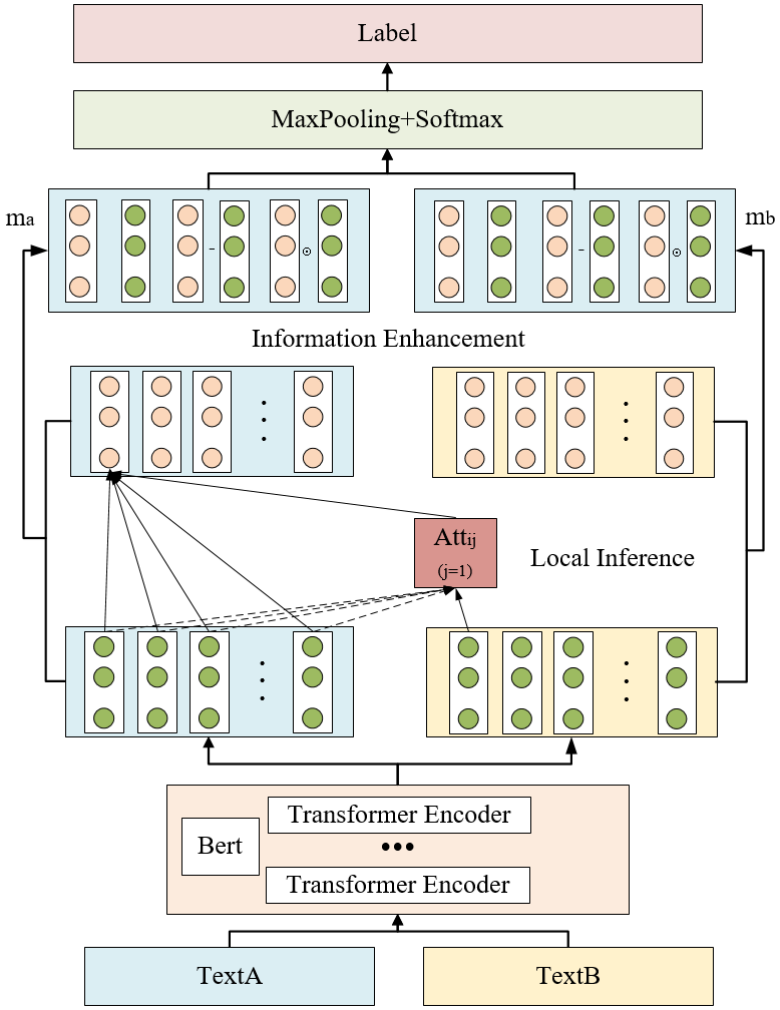

Figure 4: The model of EI-Bert

The content in $\{\bar{b}_j\}_{j=1}^{l_b}$ will be selected and represented as $\tilde{a}_i$. Similarly, the content in $\{\bar{a}_j\}_{j=1}^{l_a}$ will be selected and represented as $\tilde{b}_i$. We call this process Local Inference.

We obtain $\tilde{a}_i$ and $\tilde{b}_i$ from local inference. After that, we stack the token vectors $\tilde{a}_i$ to form the sentence vectors $\tilde{a}$. Similarly, we apply the same stack operation to generate sentence vectors of $\bar{a}$, $\tilde{b}$, and $\bar{b}$, respectively. We conduct information enhancement to calculate the difference and element-wise product of $\tilde{a}$ and $\bar{a}$, respectively. We then concatenate all the information together. The formula is as follows:

$$m_a = [\bar{a}; \tilde{a}; \bar{a} - \tilde{a}; \bar{a} \odot \tilde{a}], \tag{5}$$

$$m_b = [\bar{b}; \tilde{b}; \bar{b} - \tilde{b}; \bar{b} \odot \tilde{b}]. \tag{6}$$

where $\tilde{a}$ represents $\bar{a}$'s weighted sum, $\tilde{a} - \bar{a}$ is the difference set of $\tilde{a}$ and $\bar{a}$, $\odot$ is the dot product of $\tilde{a}$ and $\bar{a}$. m_a and m_b represent the enhanced vectors of $\bar{a}$ and $\bar{b}$ respectively. However, different sentences have different number of tokens. The dimensions of the sentence vectors vary. We concatenate the two sentences' representation vectors. We then apply Max-Pooling before Softmax which yields the prediction of the label.

4 Experimental setup

The official dataset consists of 13,694 labeled data in training, development, and test. The additional training dataset and the development dataset are released afterwards while, the test dataset consisting

of 2,960 sentences. In order to make the development set with nearly the same size of the test set, we conduct 5-fold cross-validation. For each fold, we perform data augmentation on the training dataset, the development dataset remaining unchanged. Similarly, we conduct data augmentation with 5-fold cross-validation again after adding Pseudo-Label, plus three full-volume data (without division development dataset) as the training dataset, we get 13 datasets in the end. The data pre-processing stage includes abbreviation reduction, spelling correction, word stem extraction, upper and lower case letter conversion, special symbol processing and other operations, at the same time, the data with the same training sample but inconsistent labels after the conversion of upper and lower case letters are screened out. Table 1 reports the hyperparameters of each model.

Table 1: Parameter settings for different pre-trained models

Pre-trained Model	Bert	RoBerta	XLNET
learning_rate	2e-5	5e-6	5e-6
num_train_epochs	5	30	30
batch_size	32	32	32
max_seq_length	80	80	80
warmup_proportion	0.05	0.1	0.1
max-pooling	64	128	128

The advantage of the RoBerta pre-trained model is that the data during pre-training includes about 38 GB of data extracted from Reddit, because the training dataset collected original news headlines from news media posted on Reddit. During the fine-tuning process, with the increase of the number of iterations, the loss of the default learning rate is difficult to decrease, and the model is difficult to converge. The situation can be improved by reducing the learning rate and increasing the number of iterations. Table 2 reports the accuracy of RoBerta with different inputs. We can see that: (1) The Input3 of the model contains clustering information of replacement words, and its performances is better than the Input2 and Input1; (2) The improved EI-RoBerta has better performances than RoBerta due to further interactive learning of feature information.

Table 2: The results of the accuracy of different inputs of models

Strategy	ACC
RoBerta with Input1	0.5626
RoBerta with Input2	0.5817
RoBerta with Input3	0.5976
EI-RoBerta with Input3	0.6186

In this task, we also compare the performance of a single-sentence classification and sentence pair classification by adopting Bert as the pre-trained model. sentence pair classification outperforms single-sentence classification. This makes sense due to the following two reasons: (1) Bert has absorbed the knowledge of inter-sentence relationships due to its next sentence prediction mechanism, which can help the downstream task with required inter sentence relationship; (2) We utilize Transformer, which places self-attention mechanism on sentence pair beneficial for fine tuning.

5 Results

In Subtask 2, we achieve the final results of 0.64460 accuracy and 0.25411 reward score, which was ranked the fifth in the test phrases. In Table 3, we show the top 3 results from the three pre-trained model. It is shown that 5-cv-data indicates the prediction results of the model after 5-fold cross-validation but without data augmentation, 5-cv-data-augmentation indicates the development dataset remains unchanged and the training dataset with data augmentation, and the last column indicates the results after adding

Pseudo-Label data with 5 epochs based on data augmentation. From the table we can see that under the premise of 5-fold cross-validation of different strategies, data augmentation and Pseudo-Label are conducive to model performance improvement while RoBerta attaining the best performance among the pre-trained models.

Table 3: Comparison of different strategies

Models	5-cv-data	5-cv-data-augmentation	5-cv-data-augmentation+Pseudo-Label
Bert(ACC)	0.6386	0.6398	0.6409
EI-Bert(ACC)	0.6424	0.6444	0.6434
EI-XLNET(ACC)	0.6451	0.6487	0.6490
EI-RoBerta(ACC)	0.6438	0.6470	0.6461

In the end, we select 21 models whose development set score was greater than 0.64 to predict the logits predicted on the test set for weighted summation. The weight coefficient ratio of Bert, XLNET and RoBerta is 3:4:3. In addition, the predicted results screen out 13 groups of contradictions (funniness A>B,A<C,B>C or A<B,A>C,B<C) through the discovery, because the system calculates the final evaluation results ignore the predicted results of two edited headlines with the same funniness sentences, so also filtered out 18 data predicted to be label 0, using individual model results to calculate the plurality for label correction.

6 Conclusion and Future Work

This paper presents a method to detect humor in edited news headlines of the Enhancement Inference Bert model based on auxiliary sentences. We first creatively make a variety of differently auxiliary sentence pair inputs, and then use the Bert pre-trained model to encode the sentence pair. Secondly, we perform the enhancement of local inference information on the sentence pair features. while utilizing we make data augmentation and Pseudo-Label. Finally, we carry out a multi-model weighted ensemble strategy to enhance the model performance.

In the future, we will exploit multi-task learning to simultaneously learn Subtask 1, a regression task, and Subtask 2, a classification task. By sharing the information among the tasks, we can gain more knowledge and apply a more precise data representation.

References

Francesco Barbieri, Horacio Saggion. 2014. *Automatic detection of irony and humour in twitter*, Pages 2637-2642. Proceedings of the 5th International Conference on Computational Creativity.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei. Hui Jiang, Diana Inkpen. 2017. *Enhanced LSTM for Natural Language Inference*. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, Kristina Toutanova. 2019. *BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions.* Association for Computational Linguistics.

Dmitry Davidov, Oren Tsur, Ari Rappoport. 2010. *Semi-supervised recognition of sarcastic sentences in twitter and amazon*, Pages 107–116. Proceedings of the Fourteenth Conference on Computational Natural Language Learning.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2018. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Association for Computational Linguistics.

Nabil Hossain, John Krumm, Lucy Vanderwende, EricHorvitz, Henry Kautz. 2017. *Filling the Blanks (hint: plural noun) for Mad Libs Humor*, Pages 638-647. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.

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.

Nabil Hossain, John Krumm, Michael Gamon. 2019. *"President Vows to Cut <Taxes> Hair": Dataset and Analysis of Creative Text Editing for Humorous Headlines*, Volume 1, Pages 133–142. Association for Computational Linguistics.

Mikhail Khodak, Nikunj Saunshi and Kiran Vodrahalli. 2018. *A large self-annotated corpus for sarcasm.* Language Resources and Evaluation Conference.

Dong-Hyun Lee. 2013. *Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks.* ICML 2013 Workshop : Challenges in Representation Learning.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. 2019. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. Association for Computational Linguistics.

David Matthews. 2013. *Unsupervised joke generation from big data*, Pages 228-232. Association for Computational Linguistics.

Yuchen Qiao, Haixia Wang, Dongsheng Wang. 2017. *Parallelizing and optimizing overlapping community detection with speaker-listener Label Propagation Algorithm on multi-core architecture*, Pages 439-443. 2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, Zhiyuan Liu. 2019. *Understanding the Behaviors of BERT in Ranking*. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, Richard Socher. 2019. *Unifying Question Answering, Text Classification, and Regression via Span Extraction*. Association for Computational Linguistics.

Chi Sun, Luyao Huang, Xipeng Qiu. 2019. *Utilizing Bert for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence*, Volume 1, Pages 380–385. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le. 2019. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. Association for Computational Linguistics.