MagicPai at SemEval-2021 Task 7: Method for Detecting and Rating Humor Based on Multi-Task Adversarial Training

Jian Ma, Shuyi Xie, Haiqin Yang§, Lianxin Jiang,
Mengyuan Zhou, Xiaoyi Ruan, Yang Mo
Ping An Life Insurance, Ltd.
Shenzhen, Guangdong province, China
{MAJIAN446,XIESHUYI542,JIANGLIANXIN769,RUANXIAOYI687,MOYANG853}@pingan.com.cn
§ the corresponding author, email: hqyang@ieee.org

Abstract

This paper describes MagicPai's system for SemEval 2021 Task 7, HaHackathon: Detecting and Rating Humor and Offense. This task aims to detect whether the text is humorous and how humorous it is. There are four subtasks in the competition. In this paper, we mainly present our solution, a multi-task learning model based on adversarial examples, for task 1a and 1b. More specifically, we first vectorize the cleaned dataset and add the perturbation to obtain more robust embedding representations. We then correct the loss via the confidence level. Finally, we perform interactive joint learning on multiple tasks to capture the relationship between whether the text is humorous and how humorous it is. The final result shows the effectiveness of our system.

1 Introduction

Humor is the tendency of experiences to provoke laughter and provide amusement. Regardless of gender, age or cultural background, it is a special way of language expression to provide an active atmosphere or resolve embarrassment in life while being an important medium for maintaining mental health (Lefcourt and Martin, 1986). Recently, with the rapid development of artificial intelligence, it becomes one of the most hot research topics in natural language processing to recognize humor (Nijholt et al., 2003). The task of humor recognition consists of two subtasks: whether the text contains humorous and what level of the humor it is. Early humor recognition methods tackle this task mainly by designing heuristic humor-specific features on classification models (Khodak et al., 2018) and have proved that this automatic way can attain satisfactory performance. Nowadays, researchers try to resolve this task by statistical machine learning or deep learning technologies.

The SemEval 2021 Task 7, HaHackathon: Detecting and Rating Humor and Offense, consists of four subtasks: Subtask 1 simulates the previous humor detection task, in which all scores are averaged to provide an average classification score. Subtask 1a is a binary classification task to detect whether the text is humorous. Subtask 1b is a regression task to predict how humorous it is for ordinary users in a value range from 0 to 5. Subtask 1c is also a binary classification task to predict whether the humor grade causes controversy if the text is classified as humorous. Subtask 2 aims to predict how offensive text for an ordinary user is in an integral value range from 0 and 5.

Due to the highly subjective nature of humor detection, the data is labeled by people with different profile in gender, age group, political position, income level, social status, etc. The tasks are extremely challenging because they lack a unified standard to define humor.

To tackle the tasks, we first preprocess the text, including stemming, acronym reduction, etc. We then apply the pre-trained language model to get the representation of each subword in the text as the model input. Meanwhile, we add a perturbation to the embedding layer and design an optimization goal that maximizes the perturbation of the loss function. After that, we perform interactive multi-task learning on judging whether humor exists and predicting how humorous it is. That is, based on maximizing the likelihood estimation under the Gaussian distribution with the same variance, we construct a multi-task loss function and automatically select different loss weights in the learning to improve the accuracy of each task.

2 Related Work

The early stages of humor recognition are based on statistical machine learning methods. For example,

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1153–1159
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

Taylor and Mazlack (2004) try to learn statistical patterns of text in N-grams and provide a heuristic focus for a location of where wordplay may or may not occur. Mihalcea and Strapparava (2005) show that automatic classification techniques can be effectively deploy to distinguish between humorous and non-humorous texts and obtain significant improvement over the Apriori algorithm, a well-known baseline. In addition, three human-centric features are designed for recognizing humor in the curated one-liner dataset. Mikolov et al. (2011) apply SVM models for humor recognition as a binary classification task and prove that the technique of metaphorical mapping can be generalized to identify other types of double entendre and other forms of humor. Kiddon and Brun (2011) present several modifications of the original recurrent neural network language model to solve the humor recognition task. Castro et al. (2016) collect a crowd-sourced corpus for humor classification from Spanish tweets and conduct extensive experiments to compare various machine learning models, such as Support Vector Machine (SVM), a Multinomial version of Naïve Bayes (MNB), Decision Trees (DT), k Nearest Neighbors (kNN), and a Gaussian version of Naïve Bayes (GNB). Yan and Pedersen (2017) observe that bigram language models performed slightly better than trigram models and there is some evidence that neural network models can outperform standard back-off N-gram models. Chen and Soo (2018) extend the techniques of automatic humor recognition to different types of humor as well as different languages in both English and Chinese and proposed a deep learning CNN architecture with high way networks that can learn to distinguish between humorous and nonhumorous texts based on a large scale of balanced positive and negative dataset.

With the rapid development of deep learning technology, various pre-training models have made great progress in the field of natural language processing (Yang and Shen, 2021; Wang et al., 2021; Yang et al., 2021). Liu et al. (2018) propose to model sentiment association between elementary discourse units and compare various CNN methods of humor recognition. Weller and Seppi (2019) employ a Transformer architecture for its advantages in learning from sentence context and demonstrate the effectiveness of this approach and show results that are comparable to human performance. Ma et al. (2020) propose a new algorithm Enhancement Inference BERT (EI-BERT) that performs well in sentence classification. Fan et al. (2020) propose an internal and external attention neural network (IEANN) Attention mechanism (Fan et al., 2020; Jiao et al., 2019) has been applied and show good model performance. The existing work can be borrowed or inspired our proposal in this paper.

3 Overview

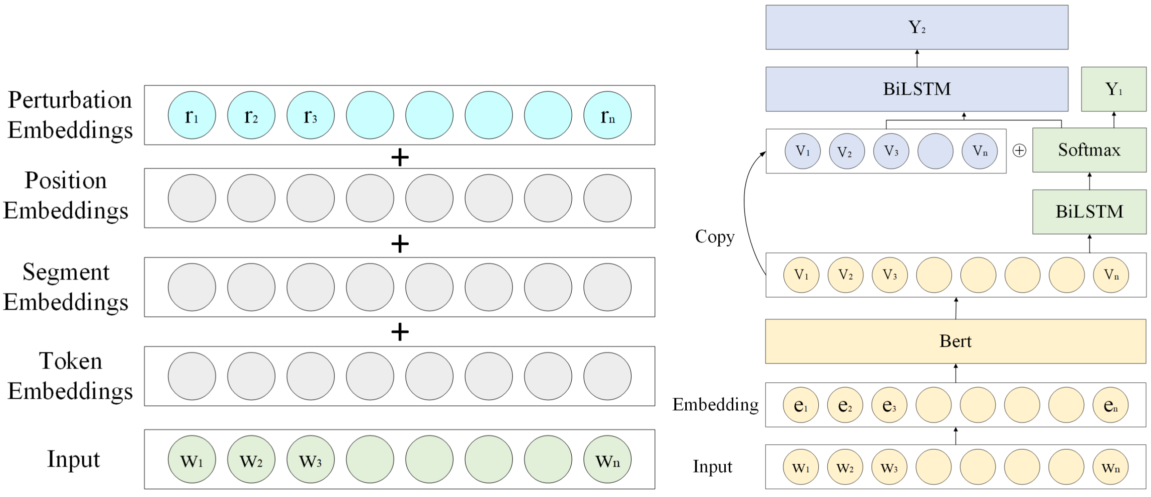
In the following, we present the implementation of our system for the competition.

3.1 Virtual Adversarial Training Based on Loss Correction

Recently, adversarial examples (Szegedy et al., 2014) have been generated to increase the robustness of training deep learning models (Pan et al., 2019; Lei et al., 2020). This work is motivated by the significant discontinuities between the input-output mappings of deep neural networks. When an imperceptible perturbation is added to the input, it may make the original normal network misclassify the result. The characteristics of these perturbations are not random artifacts of learning generated by the network during the learning process, because the same perturbation will cause different networks trained on different data sets to produce the same classification errors. Adversarial examples are samples that significantly improve the loss of the model by adding small perturbations to the input samples.

The adversarial training (Miyato et al., 2017) is a training process that can effectively identify the original sample and the adversarial sample model. Usually, the adversarial training requires labeled samples to provide supervision loss because the perturbation is designed to increase the model loss function. Virtual adversarial training (Liu et al., 2020) extends the adversarial training to semi-supervised mode by adding regularization to the model so that the output distribution of a sample is the same as the output distribution after perturbation while attaining good performance in both supervised and unsupervised tasks. When the training sample is mixed with noise, it is easy to overfit the model and learn wrong information. Therefore, it is necessary to interfere to control the influence of noise.

Figure 1(a) illustrates the perturbation in our implementation. For a word with a sequence length of n, we let w_i denote the i-th subword, where $i = 1, \ldots, n$. The representation of w_i is then



(a) Perturbation embeddings (b) Interactive multi-task learning

Figure 1: The main implementation of our proposed system.

computed by the sum of token embedding, segmentation embedding, position embedding, and perturbation embedding, an additional embedding. This makes it slightly different from the existing pre-trained language models, e.g., BERT.

The virtual adversarial training can be unified by the following objective:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}[\alpha \max_{\beta} L(f(x+\beta;\theta), f(x;\theta)) \\ + L(f(x;\theta), y)], \qquad (1)$$

where D is a training dataset consisting of input-output pairs (x, y), α is a hyperparameter to control the trade-off between the standard error and the robust error. β is the adversarial perturbation, y represents the true label, θ is the model parameter, L is the loss function. $x + \beta$ quantifies the perturbation β injecting into x. The goal of β is to maximize the difference between the two decision function values, $f(x + \beta; \theta)$ and $f(x; \theta)$, i.e., to make the prediction of the existing model as incorrect as possible. To make β meet a certain constraint, a conventional setting is to let $\|\beta\| \leq \epsilon$, where ϵ is a constant. After constructing an adversarial sample $x + \beta$ for each sample, Eq. (1) tries to seek the model parameter θ by minimizing the prediction loss.

Since the training samples are mixed with noise, it is easy for the model to overfit and learn wrong information (Reed et al., 2015), interference is adopted to control the influence of noise. The loss function is defined as follows:

$$\mathcal{L} = -\sum_{i=1}^{N}((1-w_i)y_i + w_i\tilde{y}_i)\log(l_i) \qquad (2)$$

where y_i is the true label, $\tilde{y}_i$ is the predicted label, and l_i is the predicted probability distribution. w_i is a hyperparameter to control the trade-off between true label and predicted label. By minimizing the loss defined in Eq. (2), we can reduce the attention to noise points by adding the model's own predictions to the true labels and the prediction to the noise point.

3.2 Interactive Multi-task Training

According to the description of the first two tasks, task 1a is a binary classification task to predict if the text would be considered humorous for an average user while task 1b is a regression task to determine how humorous it is for an average user when the text is classed as humorous, where the values vary between 0 and 5. In order to capture the relationship between whether text is humorous and how humorous it is, we designed the network structure shown in Fig. 1(b). The input, as illustrated in Fig. 1(a), is the sum of the token embedding, position embedding, segment embedding, and perturbation embedding. The sum of four embeddings is sent to a pre=trained language model (PLM) to yield an input for a BiLSTM model. After that, a Softmax layer is placed to recognize whether the text is humor. Meanwhile, the output of the PLM and the output of the Softmax layer are concatenated together and sent to another BiSLTM model to predict how humorous it is. In Fig. 1(b), the notation $\oplus$ represents the concatenation operation. Because two tasks have different noise patterns, learning two tasks simultaneously can make features interact in the tasks. For task 1a, it is easy to learn some important features while for task 1b, it is difficult to extract them. The reason may come from the following facts: the interaction between task 1b and the features may be too complicated, or some other features may hinder the learning procedure (Xia and Ding, 2019). Hence, by deploying interactive multi-task learning, we can get a more generalized representation.

Since different loss functions have different scales, loss functions with a larger scale will significantly dominate the loss functions with a smaller scale (Liang et al., 2020; Zhang et al., 2020). Therefore, a weighted summation of the loss function is required to make balance on the loss functions. Motivating by (Kendall et al., 2017) that modeling is based on task-dependent and homoscedastic aleatoric uncertainty, i.e., for a certain sample, the model not only predicting its label but also estimating the task-dependent homoscedastic uncertainty, we present a multi-task loss function derived by maximizing the Gaussian likelihood of the same variance uncertainty. Suppose the input is X, the parameter matrix W is the model parameter for the output, $f^W(x)$. For the classification in task 1a, the Softmax likelihood can be defined by:

$$p(y_1|f^W(x)) = Softmax(f^W(x), \sigma_1), \quad (3)$$

where σ_1 is the observed noise scalar for the classification model.

For the regression task in task 1b, we can define its probability as the Gaussian likelihood by:

$$p(y_2|f^W(x)) = G(f^W(x), \sigma_2), \quad (4)$$

where σ_2 is the observed noise scalar for the regression model.

Here, to learn the models in the multi-task mode, we define the multivariate probability by

$$\begin{aligned} &p(y1, y2|f^W(x)) && (5)\\ =&p(y_1|f^W(x)) \cdot p(y_2|f^W(x))\\ =&Softmax(f^W(x), \sigma_1) \cdot G(f^W(x), \sigma_2). \end{aligned}$$

Maximizing the probability defined in Eq. (5) is equivalent to minimizing the following objective:

$$\begin{aligned} &L(W, \sigma_1, \sigma_2) && (6)\\ =&-\log p(y1, y2|f^W(x))\\ =&-\log Softmax(f^W(x), \sigma_1) \cdot G(f^W(x), \sigma_2)\\ \propto&\frac{1}{\sigma_1^2}L_1(W) + \frac{1}{2\sigma_2^2}L_2(W) + \log\sigma_1 + \log\sigma_2 \end{aligned}$$

where $L_1 = -\log Softmax(f^W(x), y_1)$ defines the cross entropy loss between the prediction and y_1. $L_2 = \|y_2 - f^W(x)\|^2$ defines the Euclidean loss between the prediction and y_2. By minimizing the above objective, we can learn the parameters of W, σ_1, and σ_2 accordingly.

	Train			**Dev**			**Test**		
	No.	R.	L.	No.	R.	L.	No.	R.	L.
1a	8000	7:3	20	1000	5:3	19	1000	*	23
1b	4935	*	19	1000	*	19	1000	*	23
1c	4935	1:1	19	1000	1:1	19	1000	*	23
2	8000	*	20	1000	*	19	1000	*	23

Table 1: Data Statistics. R.: The ratio of positive and negative samples. L.: the average length. * indicates the data is unavailable.

	BERT	**RoBERTa**	**XLNet**	**ERNIE**
lr	2e-5	5e-6	5e-6	3e-5
nte	5	10	10	15
bs	64	32	32	32
msl	128	100	80	80
wp	0.05	0.1	0.05	0.05

Table 2: Parameters for different pre-trained language models. lr: learning rate. nte: no. of training epochs. bs: batch size. msl: max. sequence length. wp: warmup proportion.

4 Experiments

In the following, we present the data, experimental setup and analyze the results.

4.1 Data and Experimental Setup

The data is collected from the official release in (Meaney et al., 2021). We preprocess the data by spelling correction, stemming, handling special symbols, and converting all letters to lowercase, etc. Finally, we obtain the data and report the statistics in Table 1.

In the experiment, we choose the large version of four popular pre-training language models, i.e., BERT, XLNet, RoBERTa, and ERNIE. The hyperparameters of each model are tuned based on our experience and shown in Table 2. To train a good classifier, we deliver the following procedure: 1) conducting five-fold cross-validation on the training set and obtaining 20 models; 2) applying the 20 models to get the pseudo-labels of the data in the test set and extracting the data with high confidence, i.e., the predicted label score greater than 0.95 or smaller than 0.05, as new training data; 3) the pseudo label data from the test set are mixed with the original training set to train new models. Finally, 892 pseudo label data are selected and mixed with the training set to train the final models. The regression model is jointly trained with the classification models. The models that performed well in cross-validation are selected and averaged by the weighted fusion based on the confidence.

Models	AT	LC	AT + LC
BERT	0.9459	0.9490	**0.9534**
RoBERTa	0.9480	0.9482	**0.9569**
XLNet	0.9462	**0.9487**	0.9470
ERNIE	0.9491	0.9499	**0.9512**

Table 3: The performance (accuracy) of task 1a with different training strategies.

Models	1a (Acc.)	1a (F1)	1b (RMSE)
ST	0.9569	0.9470	0.6059
MT	0.9577	0.9480	0.5823
MT+WL	**0.9637**	**0.9550**	**0.5701**

Table 4: Comparison of different strategies. ST: single task. MT: multi-task. WL: weigh loss.

4.2 Results

In order to prove the effectiveness of adversarial training (AT) and loss correction (LC), we verify task 1a on four pre-training models. AT denotes the models through adversarial training by adding perturbations in the embedding layer. LC denotes the strategy to make correction on the classification cross entropy to interfere with the influence of noise on the model. AT+LC means to apply both strategies in the training. Results reported in Table 4 show that by employing individual strategy, the models can attain good performance on task 1a while employing both strategies can gain better accuracy in BERT, RoBERTa, and ERNIE.

Moreover, we verify the effectiveness of the interactive multi-task training strategy on RoBERTa. MT+WL denotes that the weighted hyperparameters in the loss function are adjusted based on uncertainty, determined by the learned σ_1's and σ_2, during interactive multi-task training to scale the output the loss function of each task in a similar range. Results reported Table 4 show that the multi-task joint training mechanism can reduce the RMSE of the regression task (i.e., 1b) significantly while adjusting the loss weight can further decrease the error.

Finally, we attain the F1 score of 0.9570 and the accuracy of 0.9653 on task 1a, respectively. The RMSE on task 1b is 0.5572. The RMSE on task 2 is 0.446.

5 Conclusion and Future Work

This paper presents our system for SemEval-2021 task 7. Several techniques, such as interactive multi-task joint training, adversarial training, and loss correction, are applied to tackle the task. More specifically, the perturbation is first added to the input embedding layer and the predicted labels are also added with the real labels to reduce the loss of the noise point data. Next, the output of task 1a by the Softmax is concatenated with the input of the task 1b to perform joint training on both tasks. Meanwhile, the uncertainty weighting scheme on the loss allows the simple task to have a higher weight. Finally, multiple models are ensembled to yield the final prediction results. Our system attains the first place in the competition.

In the future, we can explore and verify three other effective strategies. The first strategy is the task-adaptive funetuning on the pre-trained language models. Relevant sentences can be continuously fed into the pre-trained language models to improve the model performance. The second strategy is to build a graph neural network (GNN) model to exploit all vocabulary for text classification. Because BERT is relatively limited to capture the global information from a larger language vocabulary, it is promising to facilitate the GNN, which captures the global information, with the in-depth interaction of BERT's middle layers, which embed sufficient local information. We will further investigate discourse structures (Lei et al., 2017, 2018) for humor detection. Because, both BERT and GNN models information from word relations, it is necessary to involve the study of discourse structures, which describe how two sentences are logically connected to one another. By such novel design, we can attain better representations and improve the classification performance.

References

Santiago Castro, Matías Cubero, Diego Garat, and Guillermo Moncecchi. 2016. Is this a joke? detecting humor in spanish tweets. In *Advances in Artificial Intelligence - IBERAMIA 2016*, pages 139–150, Cham. Springer International Publishing.

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. Association for Computational Linguistics.

Xiaochao Fan, Hongfei Lin, Liang Yang, Yufeng Diao, Chen Shen, Yonghe Chu, and Yanbo Zou. 2020. Humor detection via an internal and external neural network. *Neurocomputing*, 394:105–111.

Wenxiang Jiao, Haiqin Yang, Irwin King, and Michael R. Lyu. 2019. Higru: Hierarchical gated recurrent units for utterance-level emotion recognition. In *NAACL-HLT*, pages 397–406. Association for Computational Linguistics.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2017. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *CoRR*, abs/1705.07115.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *LREC*, Miyazaki, Japan. European Language Resources Association (ELRA).

Chloé Kiddon and Yuriy Brun. 2011. That's what she said: Double entendre identification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 89–94, Portland, Oregon, USA. Association for Computational Linguistics.

Herbert M. Lefcourt and Rod A. Martin. 1986. *Humor and Life Stress*. Springer New York.

Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2020. Conversational recommendation: Formulation, methods, and evaluation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2425–2428.

Wenqiang Lei, Xuancong Wang, Meichun Liu, Ilija Ilievski, Xiangnan He, and Min-Yen Kan. 2017. Swim: A simple word interaction model for implicit discourse relation recognition. In *IJCAI*, pages 4026–4032.

Wenqiang Lei, Yuanxin Xiang, Yuwei Wang, Qian Zhong, Meichun Liu, and Min-Yen Kan. 2018. Linguistic properties matter for implicit discourse relation recognition: Combining semantic interaction, topic continuity and attribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hongru Liang, Wenqiang Lei, Paul Yaozhu Chan, Zhenglu Yang, Maosong Sun, and Tat-Seng Chua. 2020. Pirhdy: Learning pitch-, rhythm-, and dynamics-aware embeddings for symbolic music. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 574–582.

Lizhen Liu, Donghai Zhang, and Wei Song. 2018. Modeling sentiment association in discourse for humor recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 586–591, Melbourne, Australia. Association for Computational Linguistics.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models.

Jian Ma, ShuYi Xie, Meizhi Jin, Jiang Lianxin, Mo Yang, and Jianping Shen. 2020. XSYSIGMA at SemEval-2020 task 7: Method for predicting headlines' humor based on auxiliary sentences with EI-BERT. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1077–1084, Barcelona (online). International Committee for Computational Linguistics.

J.A. Meaney, Steven R. Wilson, Luis Chiruzzo, and Walid Magdy. 2021. Semeval 2021 task7, 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*.

Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 531–538, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur. 2011. Extensions of recurrent neural network language model. In *ICASSP*, pages 5528–5531.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. *Machine Learning*.

Anton Nijholt, Oliviero Stock, Alan Dix, and John Morkes. 2003. Humor modeling in the interface. *Conference on Human Factors in Computing Systems - Proceedings*.

Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. 2019. Recent advances in neural question generation. *arXiv preprint arXiv:1905.08949*.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. *Computer Vision and Pattern Recognition*, abs/1705.07115.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. *Computer Vision and Pattern Recognition*.

Julia M. Taylor and Lawrence J. Mazlack. 2004. Computationally recognizing wordplay in jokes. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 26.

Xinyi Wang, Haiqin Yang, Liang Zhao, Yang Mo, and Jianping Shen. 2021. Refbert: Compressing bert by referencing to pre-computed representations. In *IJCNN*.

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

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012, Florence, Italy. Association for Computational Linguistics.

Xinru Yan and Ted Pedersen. 2017. Duluth at semeval-2017 task 6: Language models in humor detection. *CoRR*, abs/1704.08390.

Haiqin Yang and Jianping Shen. 2021. Emotion dynamics modeling via bert. In *IJCNN*.

Haiqin Yang, Xiaoyuan Yao, Yiqun Duan, Jianping Shen, Jie Zhong, and Kun Zhang. 2021. Progressive open-domain response generation with multiple controllable attributes. In *IJCAI*.

Yao Zhang, Xu Zhang, Jun Wang, Hongru Liang, Wenqiang Lei, Zhe Sun, Adam Jatowt, and Zhenglu Yang. 2020. Generalized relation learning with semantic correlation awareness for link prediction. *arXiv preprint arXiv:2012.11957*.