Applying Transformers and Aspect-based Sentiment Analysis approaches on Sarcasm Detection

Taha Shangipour ataei, Soroush Javdan and Behrouz Minaei-Bidgoli
Computer Engineering Department
Iran University of Science and Technology
Tehran, Iran
taha_shangipour,soroush_javdan@comp.iust.ac.ir
b_minaei@iust.ac.ir

Abstract

Sarcasm is a type of figurative language broadly adopted in social media and daily conversations. The sarcasm can ultimately alter the meaning of the sentence, which makes the opinion analysis process error-prone. In this paper, we propose to employ bidirectional encoder representations transformers (BERT), and aspect-based sentiment analysis approaches in order to extract the relation between context dialogue sequence and response and determine whether or not the response is sarcastic. The best performing method of ours obtains an F1 score of 0.73 on the Twitter dataset and 0.734 over the Reddit dataset at the second workshop on figurative language processing Shared Task 2020.

1 Introduction

We are living in the age of social media. Many consider it as a revolution. Social media creates a variety of new possibilities; for instance, today, people can express their thought with just a tap of a finger. In the twitter platform, people are twitting around 500 million tweets per day, and it is estimated that over 2.8 million comments are posted on the Reddit every single day. This vast amount of data present an enormous opportunity for businesses, and researchers alogn with a significant number of challenges. Many companies and researchers have been interested in these data to investigate the opinion, emotions, and other aspects of them.

The usage of informal language and noisy content within social media presents many difficulties toward the opinion and emotion analysis problems. One of the main challenges in this criteria is the appearance of figurative language such as sarcasm. The sarcasm can alter the meaning of the sentence ultimately, and consequently, make the opinion analysis process error-prone. For instance, criticism may use positive words to convey a negative message. In recent years there was a growing trend to address the Sarcasm Detection problem among Natural Language Processing (NLP) researchers. Many approaches tackle the Sarcasm Detection problem by considering contextual information, instead of using utterance solely. For instance, Bamman and Smith (2015) utilized author context along with the environment and audience context, and Mishra et al. (2016) used cognitive features, Ghosh et al. (2018) made use of conversational context.

The Sarcasm Detection shared task[1] is aimed to detect sarcasm based on the conversation context. Given the current utterance and conversation history, the models are expected to decide if the utterance is sarcastic or not. We test our models on the dataset from both Twitter and Reddit. Both utterance and the conversation history have been used as input. We applied the transformer-based model and adopted aspect-based sentiment analysis approaches to address the problem.

The remnant of this paper is organized as follows: Section 2 reviews related work. Section 3 describes the datasets. Section 4 explains our methodology. Section 5 shows the results of each dataset in detail. Lastly, section 6 provides conclusions and our plans for future research.

2 Related works

There have been several attempts to solve the Sarcasm Detection problem with rule-based approaches. Bharti et al. (2015) presented two rule-based classifiers for two different types of tweet structure, the first one used to detect sarcasm in tweets that have a contradiction between negative sentiment and positive situation. The second classifier applied to tweets that start with interjection

[1] https://competitions.codalab.org/competitions/22247

Proceedings of the Second Workshop on Figurative Language Processing, pages 67–71
July 9, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

words. The former classifier applied parsed-based lexicon generation to identify phrases that display sentiment, and indicate the sarcastic label whenever a negative phrase occurs in a positive sentence. The latest classifier used interjections and intensifiers that occur together in order to predict sarcasm. Maynard and Greenwood (2014) suggests that hashtag sentiment is an essential symbol of sarcasm, and authors often used hashtags to emphasize sarcasm. They propose the tweet is sarcastic whenever the hashtags' sentiments do not agree with the rest of the tweet.

Besides the requirement for in-depth knowledge of the domain and much manual work, rule-based methods are usually not the best performers in terms of prediction quality. Because of the high cost of the rule-based methods, many researchers put there focus on machine learning approaches. Different types of models and features have been adopted to tackle this problem. Mukherjee and Bala (2017) addressed the problem in both supervised and unsupervised settings. They utilized Naïve Bayes as a classifier and C-means clustering, which is one of the most widely used fuzzy clustering algorithms. Joshi et al. (2016) adopted a sequence labeling techniques (SVM-HMM and SEARN) and indicated that sequence labeling algorithms outperform the classification algorithms in conversational data.

With the development of computational hardware and deep learning in recent years, many deep learning methods have been proposed to address the sarcasm detection problem. Amir et al. (2016) proposed a Convolutional Neural Network-based architecture that jointly learns and exploits embeddings for the users' content and utterances. Ghosh and Veale (2016) used the complication of the Convolutional Neural Network and Recurrent Neural Network. They used two layers of Convolutional Neural Network, followed by two layers of Long Short-Term Memory(LSTM). The output of LSTM layers fed to a Fully Connected Neural Network in order to produce a higher-order feature set. Diao et al. (2020) proposed a novel multi-dimension question answering network in order to detect sarcasm. They utilized conversation context information. A deep memory network based on BiLSTM and attention mechanisms have been adopted to extract the factors of sarcasm.

3 Dataset

Two corpora were used in Sarcasm Detection shared task, which both of them are balanced. The Twitter corpus consists of 5000 data samples for the train and 1800 for the test set. On the other hand, Reddit corpus contains 4400 data samples for the train and 1800 for the test set. Training datasets have four columns:

- ID: a unique identifier for each data sample
- Context: an ordered list of dialogues
- Response: reply to the last post or tweet of Context dialogues
- Label: indicate wheter the responce is sarcastic or not

Figure 1 shows the distribution of dialogue turns in each dataset.

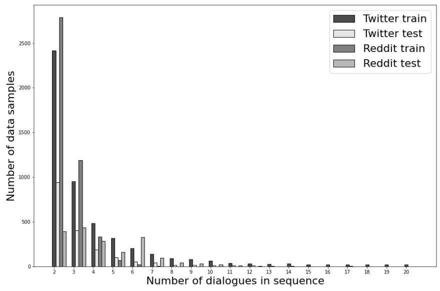

Figure 1: distribution of dialogues turns.

Moreover, we used a balanced dataset proposed at (Khodak et al., 2017) with 1 million data samples over Reddit comments as an additional dataset.

4 Methodology

In this section we describe models and technquies that we used to address sarcasm detection problem.

4.1 Preprocessing

Hashtag segmentation: the hashtag is a type of metadata used on social media starting with a number sign, #, which helps users find messages with the same topic. We apply word segmentation on hashtags, for example '#BlackHistoryMonth' is segmented as 'Black History Month.'

Misc.: We removed all of the @USER mentions and <URL> tags.

4.2 Word Embedding:

GloVe (Pennington et al., 2014) is an unsupervised method for extracting word vector representation for our raw data. We also employed Fasttext(Mikolov et al., 2018) embedding because they are derived from character n-gram and thus are useful for misspelled words and social media contents.

4.3 Models:

NBSVM: We used the NBSVM model introduced by Wang and Manning (2012), which is a combination of Naïve Bayes and support vector machine, and is known as an excellent baseline for many NLP tasks. As input, we utilized the TF-IDF matrix with character n-gram features with n-gram range from 2 to 6. We applied this method over both datasets. Also, we tried different input data for the NBSVM model. The different combinations of 'response' column, 'context' column have been used as input.

BERT: Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2018) was released by the Google research team and achieved state of the art in many NLP tasks. BERT is pretrained on a huge corpus of data sources. As input, we experiment with the 'response' column solely, 'context' column solely, and the concatenation of 'context' and 'response' column. We trained the model with three epochs, a batch size of 8, and a learning rate of 2e-5. For maximum sequence length, the 128 yield best result.

BERT-SVM: We used logits from the final layer of BERT as input for a support vector machine model with a linear kernel. We trained the model with three epochs, a batch size of 8, and a learning rate of 2e-5. For maximum sequence length, the 128 yield best result.

BERT-LR: We used logits from the final layer of BERT as input for a logistic regression model. We trained the model with three epochs, a batch size of 8, and a learning rate of 2e-5. For maximum sequence length, the 128 yield best result.

XLNET: XLNet (Yang et al., 2019) is a generalized autoregressive pretraining method. Since it outperforms BERT on 20 different NLP tasks, we train this method over the Reddit dataset. We trained the model with three epochs, a batch size of 8, and a learning rate of 2e-5. For maximum sequence length, the 128 yield best result.

Bi-GRU-CNN+BiLSTM-CNN: CNN is suitable for detecting patterns, and by changing kernel sizes, it also can detect different patterns regardless of their positions. RNN is a sequence of network blocks linked to each other, and each of them passes a message to the next one, this feature enables the network to demonstrate dynamic temporal behavior for a time sequence. We employ a neural network architecture built on top of a concatenation of glove embedding and the fastText embedding, both of them with 300 dimensions. Then, the network splits into two parallel parts. The first part combines a bidirectional gated recurrent unit (GRU) with 128 hidden units and a convolutional layer with a kernel size of 2 and 64 hidden units. The second part combines a BiLSTM with 128 hidden units and a convolutional layer with a kernel size of 2 and 64 hidden units. Finally, we concatenate global max pooling and global average pooling of parallel parts and feed them to a dense layer and then through a softmax layer for the classification purpose.

After reviewing some aspect-based sentiment analysis methods, we found some similarities between these models and sarcasm detection problems, so we attempted to change aspect-based sentiment analysis and adapt it sufficiently to address the Sarcasm Detection problem. For all the following models, the number of training epochs has been set to 10.

IAN: IAN (Ma et al., 2017) has two attention layers that learn context and target interactively and make representation for both separately. We replace the context of the ABSA with the last dialogue in the 'context' column of the sarcasm datasets and target with the 'response' column. We utilized 300 hidden units for both LSTM and attention parts. We run this method on both datasets.

LCF-BERT: LCF-BERT (Zeng et al., 2019) is a method based on multi-head self-attention, it employs context features dynamic mask (CDM) and context features dynamic weighted (CDW) layers along with a BERT-shared layer to extract long-term internal dependencies of local context and global context in aspect-based sentiment classification problem. We alter the model input so it can perform on the sarcastic dataset. As input, we used 'response' and the last dialogue in the 'context' column. The BERT-base-uncased with a maximum sequence length of 80 has been used as a BERT-shared layer.

BERT-AEN: AEN-BERT (Song et al., 2019) is another method proposed for the aspect-based sentiment classification that we borrowed for this task. This method introduces an attentional encoder network as a solution for the RNN problem with long-term pattern recognition. It also applies a BERT-base-uncased pre-trained model with a maximum sequence length of 80. We also used hidden units of 300 for the attention part. We modify the model input so it can work on the sarcastic dataset. As input, we used 'response' and the last dialogue in the 'context' column.

5 Results

On twitter corpus, the performance of NBSVM as a simple model is quite impressive. As long as the data is from social media and might contain informal and misspelling content using character n-gram TFIDT matrix can yield excellent performance. As features, we used different combinations of 'response' column, 'context' column. However, taking the 'response' column as a feature solely produced the best result. We also test models with and without preprocessing steps. However, adding the preprocessing step did not show a significate change in the results. As expected, BERT achieved the second-best position on the scoreboard. We used a different set of features, but again using the 'response' column solely scored the best among others. Furthermore, LCF-BERT, which is an aspect-based sentiment classification method, scored the best on the Twitter dataset because aspect-based sentiment classification methods consider input data as two different sections and try to learn them interactively. The complete results with more details are shown in Table. 1.

Method	**Score**
Base-line	
NBSVM	0.691
Transformers	
BERT-base-cased	0.726
Bi-GRU-CNN+BiLSTM-CNN	0.660
Aspect-based	
LCF-BERT	**0.730**
IAN	0.648
BERT-AEN	0.651

Table 1: Models performance over Twitter dataset

Unlike our experience on the Twitter dataset, NBSVM did not perform well on the Reddit dataset. It appears that the Reddit dataset is more complicated and challenging than twitter. However, using an additional dataset, around 1 million data points, boosted the NBSVM result around 7 percent. For NBSVM, we used the same feature as it was used for the twitter dataset. BERT performance was the best on this dataset, regardless of additional data. For BERT-SVM and BERT-LR, we only utilized the 'response' column as input. Moreover, for XLNet, we used the 'response' column with 100,000 random data points from the additional dataset. Furthermore, for the aspect-based sentiment analysis models, we used the last dialogue 'context' column and 'response' column as our input. The complete results with more details are shown in Table. 2.

Method	**Score**
Base-line	
NBSVM	0.675
Transformers	
BERT-base-cased	**0.734**
BERT-SVM	0.647
BERT-LR	0.649
Bi-GRU-CNN+BiLSTM-CNN	0.660
XLNet	0.698
Aspect-based	
IAN	0.502
BERT-AEN	0.612

Table 2: Models performance over Reddit dataset

6 Conclusion

Our proposed methods ranked 5 out of 37 groups for the Reddit dataset and ranked 25 out of 36 for the Twitter dataset. This result shows the strength of the BERT pre-trained model on sarcasm detection and its combination with aspect-based sentiment analysis models, which take data as two separate parts and learn them interactively. Also, additional data can improve performance slightly better. It is noteworthy to mention that NBSVM performance as a simple baseline with the TFIDF matrix with character n-gram was quite impressive. For future work, a combination of contextual and character-based embedding could lead to better performance. Moreover, since social media content usually contains misspelling and informal data, more complicated preprocessing techniques like social media content normalization might be more helpful than proposed techniques.

References

Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

David Bamman and Noah A Smith. 2015. Contextualized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media*.

Santosh Kumar Bharti, Korra Sathya Babu, and Sanjay Kumar Jena. 2015. Parsing-based sarcasm sentiment recognition in twitter data. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1373–1380. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yufeng Diao, Hongfei Lin, Liang Yang, Xiaochao Fan, Yonghe Chu, Kan Xu, and Di Wu. 2020. A multi-dimension question answering network for sarcasm detection. *IEEE Access*.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 161–169.

Debanjan Ghosh, Alexander R Fabbri, and Smaranda Muresan. 2018. Sarcasm analysis using conversation context. *Computational Linguistics*, 44(4):755–792.

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, and Mark Carman. 2016. Harnessing sequence labeling for sarcasm detection in dialogue from tv series 'friends'. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 146–155.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579*.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. *arXiv preprint arXiv:1709.00893*.

Diana G Maynard and Mark A Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *LREC 2014 Proceedings*. ELRA.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016. Harnessing cognitive features for sarcasm detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Berlin, Germany. Association for Computational Linguistics.

Shubhadeep Mukherjee and Pradip Kumar Bala. 2017. Sarcasm detection in microblogs using naïve bayes and fuzzy clustering. *Technology in Society*, 48:19–27.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *arXiv preprint arXiv:1902.09314*.

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. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Biqing Zeng, Heng Yang, Ruyang Xu, Wu Zhou, and Xuli Han. 2019. Lcf: A local context focus mechanism for aspect-based sentiment classification. *Applied Sciences*, 9(16):3389.