Transformers on Sarcasm Detection with Context

Amardeep Kumar
Indian Institute of Technology
(ISM), Dhanbad
adkr6398@gmail.com

Vivek Anand
IIIT Hyderabad, India
vivek.a@research.iiit.ac.in

Abstract

Sarcasm Detection with Context, a shared task of Second Workshop on Figurative Language Processing (co-located with ACL 2020), is study of effect of context on Sarcasm detection in conversations of Social media. We present different techniques and models, mostly based on transformer for Sarcasm Detection with Context. We extended latest pre-trained transformers like BERT, RoBERTa, spanBERT on different task objectives like single sentence classification, sentence pair classification, etc. to understand role of conversation context for sarcasm detection on Twitter conversations and conversation threads from Reddit. We also present our own architecture consisting of LSTM and Transformers to achieve the objective.

1 Introduction

With advent of Internet and Social media platforms, it is important to know actual sentiments and beliefs of its users, and recognizing Sarcasm is very important for this. We can't always decide if a sentence is sarcastic or not without knowing its context. For example, consider below two sentences S1 and S2.

S1: "What you love on weekends?"

S2: "I love going to the doctor."

Just by looking at the 'S2' sentence we can tag the sentence 'S2' as "*not sarcastic*", but imagine this sentence as a reply to the sentence 'S1' , now we would like to tag the sentence 'S2' as *"sarcastic"*. Hence it is necessary to know the context of a sentence to know sarcasm.

We were provided with conversation threads from two of popular social media, Reddit and Twitter. For this objective We used different pre-trained language model and famous transformer architecture like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and spanBERT (Joshi et al., 2020). We also propose our own architecture made of Transformers (Vaswani et al., 2017) and LSTM (Hochreiter and Schmidhuber, 1997).

2 Datasets

Two types of Datasets were used, corpus from Twitter conversations and conversation threads from Reddit.

Twitter Corpus Ghosh et al. (2018) introduced a self label twitter conversations corpus. The sarcastic tweets were collected by relying upon hashtags, like sarcasm, sarcastic, etc., that users assign to their sarcastic tweets. For non-sarcastic they adopted a methodology, according to which Non-sarcastic tweet doesn't contain sarcasm hashtag instead they were having sentiments hashtag like happy, positive, sad, etc.

Reddit Corpus Khodak et al. (2018) collected 1.5 million sarcastic statement and many of non-sarcastic statement from Reddit. They self annotated all of these Reddit corpus manually.

For both datasets, the training and testing data was provided in json format where each utterance contains the following fields: 1) "label" : SARCASM or NOT_SARCASM. For test data, label was not provided. 2) "response" : the sarcastic response, whether a sarcastic Tweet or a Reddit post. 3) "context" : the conversation context of the "response". 4) "id" : unique id to identify and label each data point in test dataset.

Twitter data set is of 5,000 English Tweets balanced between the "SARCASM" and "NOT_SARCASM" classes and Reddit dataset is of 4,400 Reddit posts balanced between the "SARCASM" and "NOT_SARCASM" classes.

3 Pre-Process

We used different text pre-processing technique to remove noise from text provided to us. We removed unwanted punctuation, multiple spaces, URL tags, etc. We changed different abbreviations to their

Proceedings of the Second Workshop on Figurative Language Processing, pages 88–92
July 9, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

proper format, for example: ”I’m” was changed to ”I am”, ”idk” to ”I don’t know”, etc.

4 Experiments

We experimented with different transformers and pretrained models like BERT , RoBERTa, spanBERT and our own architecture built over these Transformers.

For both datasets, each training and testing utterance contains two major fields: ”response” (i.e, the sarcastic response, whether a sarcastic Tweet or a Reddit post), ”context” (i.e., the conversation context of the ”response”). The ”context” is an ordered list of dialogue, i.e., if the ”context” contains three elements, ”c1”, ”c2”, ”c3”, in that order, then ”c2” is a reply to ”c1” and ”c3” is a reply to ”c2”. Further, if the sarcastic ”response” is ”r”, then ”r” is a reply to ”c3”. For instance, for the following example, ”label”: ”SARCASM”, ”response”: ”Did Kelly just call someone else messy? Baaaahaaahahahaha”, ”context”: [”X is looking a First Lady should”, ”didn’t think it was tailored enough it looked messy”]. The response tweet, ”Did Kelly...” is a reply to its immediate context ”didn’t think it was tailored...” which is a reply to ”X is looking...”.

For each utterance in datasets, We defined ’response’ as response_string and concatenation of all the ’context’ in reverse order as context_string.

response_string = ”response”

context_string = ”c3” + ”c2” + ”c1”

We approached this classification task in two ways, first as Single sentence classification task and second as Sentence pair classification tasks. We also experimented single sentence classification only with response_string. Throughout the experiment we used ’transformers’ library by Hugging Face (Wolf et al., 2019) for experimenting with BERT and RoBERTa models and for spanBERT we used their official released code, and incorporated new methods to suit our task.

4.1 Single sentence Classification Task

As name indicates, to obtain a single sentence for classification, we concatenated response_string and context_string.

Figure 1 represents general architecture of models used in subsection 4.1.1, 4.1.2 and 4.1.3, for single sentence classification where:

- Input : response_string + context_string

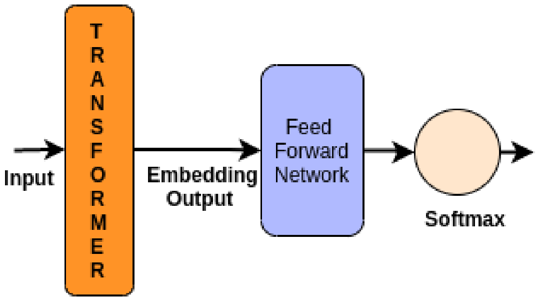

Figure 1: Transformer model for single sentence classification

- Transformer: layer could be any of the model from BERT, RoBERTa or spanBERT as transformer.
- Embedding output: is representation of ”[CLS]” token by transformer, used for classification task.
- Feed Forward Network : has multiple dense and dropout layer.
- Softmax: classifier for binary classification.

4.1.1 BERT

Devlin et al. (2019) introduced Bidirectional Encoder Representations from Transformers(BERT). BERT’s key technical innovation is applying bidirectional training of Transformers to language modeling. BERT is pre-trained on two objectives, Masked language modeling (MLM) and next sentence prediction (NSP).

We used ’bert-base-uncased’ and ’bert-large-uncased’ pretrained model in transformer layer. ’bert-base-uncased’ has 12-layers, 768-hidden state size, 12-attention heads and 110M parameters, with each hidden state of (max_seq_len, 768) size and embedding output of 768 length. ’bert-large-uncased’ has 24-layers, 1024-hidden state size, 16-attention heads and 340M parameters. it has each hidden state of (max_seq_len, 1024) size and embedding output of 1024 length. ’bert-large-uncased’ gave better results than ’bert-base-uncased’ on both datasets.

4.1.2 SpanBERT

Joshi et al. (2020) introduced pretraining method to represent and predict span instead of words. This approach is different from BERT based pretraining methods in two ways:

1. Masking contiguous random spans instead of masking random tokens.

2. Span Boundary Objective: Predicting entire content of masked span with help of hidden states of boundary token of masked span.

We used 'spanbert-base-cased' and 'spanbert-large-cased' pretrained model as transformer layer. 'spanbert-base-cased' has 12-layers, 768-hidden state size, 12-attention heads and 110M parameters, with each hidden state of (max_seq_len, 768) size and embedding output of 768 length. 'spanbert-large-cased' has 24-layers, 1024-hidden state size, 16-attention heads and 340M parameters. It has each hidden state of (max_seq_len, 1024) size and embedding output of 1024 length. 'spanbert-large-cased' gave better results than 'spanbert-base-cased', 'bert-base-uncased' and 'bert-large-uncased' respectively on both datasets .

4.1.3 RoBERTa

Liu et al. (2019) presented a replication study of BERT pre-training, related to impact of key hyper-parameter and size of training data on which it was pre-trained, and found BERT as significantly untrained.

We tried only roberta large models, which has 24-layers, 1024-hidden state size, 24-attention heads and 355M parameters. it has each hidden state of (max_seq_len, 1024) size and embedding output of 1024 length. 'roberta-large' gave better results than all previous models.

4.1.4 LSTM over Transformer

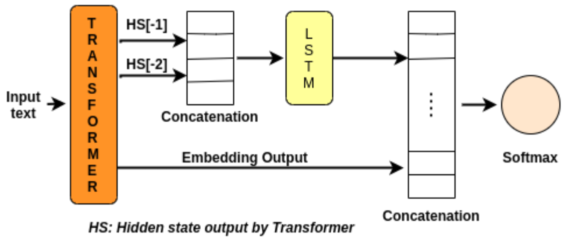

Figure 2: Architecture of model 4.1.4

To improvise, We modified previously used model architecture. Figure 2 represents architecture of our successful improvised model, where:

- HS[-1], HS[-2]: represent last hidden state and second last hidden state output by transformer respectively.
- Concatenation: layer concatenate two or more tensors along suitable axis.
- LSTM: Hochreiter and Schmidhuber (1997)

In this model, last two hidden states are concatenated and passed through LSTM to get more contextual representation of text. Later output of LSTM and embedding output of transformer is concatenated and fed through feed forward Neural network for classification.

We tried 'bert-large-uncased' and 'Roberta large' as transformer layer in this architecture. 'Roberta large' gave best f1-score among all. This model also gave best result on classification using only 'response_string' as input on both datsets.

4.2 Sentence pair Classification task

In this Sentence Pair classification task, we give a pair of text as input for binary classification. We present following two models:

4.2.1 Siamese Transformer

Our architecture was inspired from two things, first is intuition that it may be a case that only 'response' is Sarcastic but not concatenation of 'response' and 'context', and second, Siamese network (Mueller and Thyagarajan, 2016).

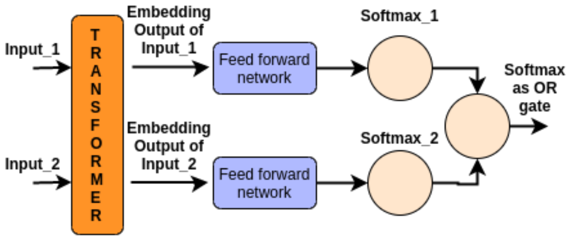

Figure 3: Architecture of Proposed Siamese Transformer

Figure 3 represents our Siamese Transformer, where: 'input_1' is response_string, 'input_2' is response_string + context_string, 'Softmax' is last softmax layer intuitively work as 'OR' logical gate.

We expected improvement in result over previous models, but it didn't happen. This also establishes that context is necessary for Sarcasm Detection.

4.2.2 Dual Transformer

Length of context_string is larger than response_string so it might be that their combined contextual representation is dominated by 'context_string'. To overcome this, we pass them through different transformers to get their individual representation of equal size. These representation are then concatenated and passed through Bi-LSTM to get contextual representation of the

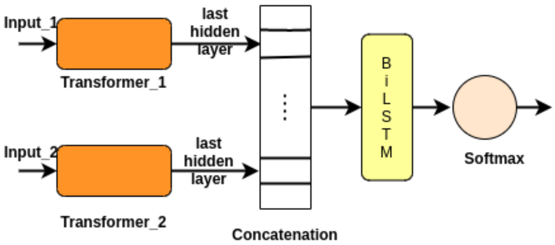

Figure 4: Architecture of Dual Transformer model

combination. Figure 4 represents our architecture of Dual transformer, where: 'input_1' is response_string, 'input_2' is context_string, 'BiLSTM' is bidirectional LSTM (Schuster and Paliwal, 1997)

Last hidden state output of both transformers are concatenated and passed over Bi-LSTM to get a better contextual, output of which is passed through a classification layer. This model didn't give better results as expected. We guessed lack of training data as one of the possible reason.

5 Results

Model	P	R	f1
response only$_{SS}$	68.7	71.8	67.50
bert-base$_{SS}$	74.7	74.8	74.62
bert-large$_{SS}$	75.1	75.1	75.03
roberta-large$_{SS}$	76.1	76.1	76.05
spanbert-base$_{SS}$	74.9	75.2	74.89
spanbert-large$_{SS}$	75.7	75.9	75.68
bert-large$_{LoT}$	76.5	76.7	76.44
roberta-large$_{LoT}$	77.3	77.4	**77.22**
roberta-large$_{ST}$	75.5	75.5	75.49
roberta-large$_{DT}$	75.9	76.2	75.88

Table 1: Result on Twitter

Model	P	R	f1
response only$_{SS}$	64.2	64.7	63.83
bert-base$_{SS}$	66.5	66.6	66.47
bert-large$_{SS}$	67.3	67.3	67.27
roberta-large$_{SS}$	67.5	67.5	67.49
spanbert-base$_{SS}$	66.9	67.3	66.75
spanbert-large$_{SS}$	67.4	67.4	67.36
bert-large$_{LoT}$	68.1	68.1	68.0
roberta-large$_{LoT}$	69.3	69.9	**69.11**
roberta-large$_{ST}$	67.9	68.1	67.86
roberta-large$_{DT}$	68.1	68.1	68.1

Table 2: Result on Reddit

Table 1 and Table 2 depict results of all models and tasks on Twitter and Reddit datasets respectively. In both table 'SS' denotes single sentence classification task, 'LoT' denotes LSTM over Transformer(4.1.4), 'DT' denotes Dual Transformer(4.2.2) and 'ST' denotes Siamese Transformer (4.2.1).

Using only 'response_string' (i.e without using context) we got best f1-score of 67.50 and 63.2 on Twitter and Reddit datsets respectively. Using response as well as context, LSTM over Transformer model (sub-section 4.1.4) with 'robert-large' as transformer layer performed best. We tried different maximum sequence legth, 126 on Twitter conversation and 80 on Reddit Conversation text gave the best results. We didn't benchmark our results with Ghosh et al. (2018), Zhang et al. (2016), etc. related works, becuase those models were trained on different datasets. To do a fair comparison, we would have to re-train those models on our dataset, but due to computational constraints we were unable to do this.

6 Related Work

Most of the existing works are on detecting sarcasm without considering context. Joshi et al. (2016) , Zhang et al. (2016) , Ghosh et al. (2018) have considered context and utterances separately for sarcasm detection and showed how context is helpful in sarcasm detection.

7 Conclusion

To conclude, we showed effective method for sarcasm detection and how much context is necessary for it. We didn't use any dataset (reddit and twitter) specific pre-processing or hyperparameter tuning in order to evaluate effectiveness of models across various types of data. In future, we would like to experiment with supplementing external data or merging different types of data on this task.

References

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. Association for Computational Linguistics.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, and Mark J. 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, Berlin, Germany. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *thirtieth AAAI conference on artificial intelligence*.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460, Osaka, Japan. The COLING 2016 Organizing Committee.