Context-Aware Sarcasm Detection Using BERT

Arup Baruah[1], Kaushik Amar Das[1], Ferdous Ahmed Barbhuiya[1], and Kuntal Dey[2]
[1]IIIT Guwahati, India
[2]IBM Research, New Delhi, India
arup.baruah@gmail.com, kaushikamardas@gmail.com,
ferdous@iiitg.ac.in, kuntadey@in.ibm.com

Abstract

In this paper, we present the results obtained by BERT, BiLSTM and SVM classifiers on the shared task on *Sarcasm Detection* held as part of *The Second Workshop on Figurative Language Processing*. The shared task required the use of conversational context to detect sarcasm. We experimented by varying the amount of context used along with the response (*response* is the text to be classified). The amount of context used includes (i) zero context, (ii) last one, two or three utterances, and (iii) all utterances. It was found that including the last utterance in the dialogue along with the response improved the performance of the classifier for the Twitter data set. On the other hand, the best performance for the Reddit data set was obtained when using only the response without any contextual information. The BERT classifier obtained F-score of 0.743 and 0.658 for the Twitter and Reddit data set respectively.

1 Introduction

Figurative language refers to texts where the intended meaning does not coincide with the literal meanings of the words and sentences that are used (Glucksberg, 2001). An example of such a sentence is *"The economic impact of Covid-19 that we have seen so far is just the tip of the iceberg"*. The *principle of compositionality* which states the meaning of a sentence can be obtained by combining the meaning of the constituent words do not apply in such sentences. Some of the types of figurative language are metaphor, idioms, similie, personification, hyperbole, understatement, analogy, irony, and sarcasm.

The Second Workshop on Figurative Language Processing, co-located with ACL 2020, had two shared tasks: *Methaphor Detection* and *Sarcasm Detection*. The shared task on sarcasm detection is a binary classification task where it is required to determine if the final response given in a conversation dialogue is sarcastic or not. So, the task was sarcasm detection given the context in which the response was made. To capture the context the full dialogue thread was provided. The task was held for two different data sets: the Twitter dataset and the Reddit dataset.

In this paper, we describe the work we performed for context aware sarcasm detection for both the data sets. We used the Bidirectional Encoder Representations from Transformers (BERT), Bidirectional Long Short-Term Memory (BiLSTM) and Support Vector Machine (SVM) classifiers in our study. The rest of this paper is structured as follows: section 2 discusses the related work that has been performed on automatic sarcasm detection, section 3 describes the data set used in this shared task, section 4 discusses the approach we used in our study, and section 6 discusses the results we obtained.

2 Related Work

Joshi et al. (2017) provides a comprehensive survey of the work performed in the field of automatic sarcasm detection. As mentioned in this survey, the use of context information beyond the target text is one of the three milestones in the research related to automatic sarcasm detection. Three types of context has been mentioned in this study: author-specific context, conversational context, and topical context. Our work in this shared task makes use of the conversational context to assist classification.

Ghosh et al. (2017) found that modeling both conversational context and response improves the F1 score for sarcasm detection by 6 to 11% compared to modeling only the response. Conditional LSTM classifiers and LSTM classifiers with attention were used in this study. Hazarika et al. (2018)

Proceedings of the Second Workshop on Figurative Language Processing, pages 83–87

https://doi.org/10.18653/v1/P17

Type	Total	POS	NEG	Max Utterances	Min Utterances	Max Length	Min Length
Train	5000	2500	2500	20	2	1213	27
Test	1800	-	-	18	2	786	24

Table 1: Twitter Dataset Statistics

Type	Total	POS	NEG	Max Utterances	Min Utterances	Max Length	Min Length
Train	4400	2200	2200	8	2	422	12
Test	1800	-	-	13	2	646	19

Table 2: Reddit Dataset Statistics

combined both content and contextual information to detect sarcasm. The contextual information captured user traits and topical information of the discussion forum. The contextual information used in Bamman and Smith (2015) consisted of author information, audience information, and the tweet against which the response is made. The contextual information was combined with content information to make the final classification. It was found that combining all the four types of features yielded the best accuracy while using only the content features resulted in the worst accuracy.

Ilic et al. (2018) used an ELMo based BiLSTM classifier to detect sarcasm and obtained superior performance on 6 out of the 7 datasets used in the study.

3 Data Set

The shared task on sarcasm detection required detecting sarcasm for two different data sets: the Twitter Dataset and the Reddit Dataset. Tables 1 and 2 show the statistics of the two data sets. As can be seen from the tables, both the train data sets are balanced with 50% of the instances labelled as *sarcasm* and 50% labelled as *not sarcasm*. The tables also list the minimum and maximum number of utterances included from the conversational dialogue apart from the response. As can be seen, the Twitter train set included from 2 to 20 utterances, the Twitter test set included from 2 to 18 utterances, the Reddit train set included from 2 to 8 utterances, and the Reddit test set included from 2 to 13 utterances. It was observed that for 48%, 52%, and 63% of the total instances in the Twitter train, Twitter test, and Reddit train data set only two utterances were present in the dialogue. In the case of the Reddit test data set, 24% of the instances had only two utterances. The rest of the instances had more than two utterances in the dialogue.

The two tables also show the minimum and the maximum length (in terms of number of tokens) of the string obtained by concatenating the response and all the utterances in the conversational dialogue. This length varied from 27 to 1213 for Twitter train data set, from 24 to 786 for Twitter test data set, from 12 to 422 for Reddit data set, and from 19 to 646 for Reddit data set. Although the maximum length is high, it was seen that 73% and 75% of the instances in the Twitter train and test had length less than 150, and 99% and 75% of the instances in Reddit train and test data set had length less than 100 respectively.

4 Methodology

This section discusses the details required for reproducing the results. It mentions the preprocessing steps, the architecture of the classifiers used, and hyperparameter values.

4.1 Preprocessing

The preprocessing performed includes the removal of the *USER* and *URL* tokens from the response and utterances in the dialogue. The text was also converted to lower-case.

4.2 Classifiers

In our study, we used BiLSTM, BERT, and SVM classifiers.

4.2.1 BiLSTM

The BiLSTM classifier (Hochreiter and Schmidhuber, 1997) we used had a single BiLSTM layer of 100 units. The output from the BiLSTM layer is fed to a fully connected layer of 100 units through a max pooling layer. After applying dropout on

the output from the fully connected layer, it was fed to an output layer having a single unit. The hyperparameter values used for the classifier are listed in table 3.

Parameter	Value
Number of LSTM units	100
LSTM dropout	0.25
Recurrent dropout	0.10
Units in 1st Dense layer	100
Activation Function for 1st Dense layer	ReLU
Rate for dropout layer	0.25
Units in 2nd Dense layer	1
Activation Function for 2nd Dense layer	sigmoid
Optimizer	Adam
Learning Rate	2e-5
Loss Function	Binary cross-entropy

Table 3: Hyperparameters for the BiLSTM model

For the BiLSTM classifier, the text was represented using the pre-trained fastText embeddings. The 300-dimensional fastText embeddings [1] trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset were used in our study.

4.2.2 BERT

BERT (Devlin et al., 2019) is a transformer based architecture (Vaswani et al., 2017). It is a bi-directional model. As opposed to static embeddings that are produced by fastText, BERT produces contextualized word embeddings where the vector for the word is computed based on the context in which it appears.

In our study, we used the uncased large version of BERT [2]. This version has 24 layers and 16 attention heads. This model generates 1024 dimensional vector for each word. We used 1024 dimensional vector of the Extract layer as the representation of the text. Our classification layer consisted of a single Dense layer. This layer used the *sigmoid* activation layer. The classifier was trained using the *Adam* optimizer with a learning rate of 2e-5. The *binary crossentropy* loss function was used.

[1] https://fasttext.cc/docs/en/english-vectors.html

[2] https://github.com/google-research/bert

4.2.3 SVM

The Support Vector Machine (SVM) classifier we used in our study was trained using the TF-IDF features of character n-grams (1 to 6). The *linear* kernel was used for the classifier and hyperparameter C was set to 1.0.

5 Experiments

In our experiments, we concatenated the response with a varied number of utterances from the dialogue. It this way we varied the amount of context that was used to detect sarcasm. While concatenating, the utterances were concatenated in the reverse order so that the last utterances appeared at the start of the string. The individual utterances were separated from each other using a special token. We performed the following types of experiments:

1. Used only the response without any context
2. Used the response along with the last utterance from the dialogue
3. Used the response along with the last two utterance from the dialogue
4. Used the response along with the last three utterance from the dialogue
5. Used the response along with all the utterances from the dialogue

6 Results and Discussion

Tables 4 and 5 shows the results that our classifiers obtained on the test data sets. The scores mentioned in the tables were obtained from the submission page of the shared task in CodaLab [3].

We cross validated our models by retaining 20% of the train set as our development set. This technique has a disadvantage that the model we submit does not see 20% of the instances from the train set. So, the other form of cross validation we performed is the 5-Fold cross validation. The development set results were used to filter out some of the models and hyperparameter values (as we did not use grid search). The implementation details of our models and the hyperparameter values used are discussed in section 4.

As can be seen from table 4, the best F-score of 0.743 was obtained by the BERT classifier for the

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

Classifier	Validation Technique	Amount of conversational context used	Max Seq Length	Precision	Recall	F1
BERT	5-fold CV	Only Response	70	0.741	0.741	0.741
BERT	5-fold CV	Response+Last Utterance	140	**0.744**	**0.748**	**0.743**
BERT	5-fold CV	Response+Last 2 Utterances	180	0.500	0.500	0.500
BERT	5-fold CV	Response+Last 3 Utterances	260	0.500	0.250	0.334
BERT	5-fold CV	Response+All Utterances	300	0.734	0.735	0.734
BERT	20% holdout	Response+All Utterances	300	0.724	0.725	0.724
BiLSTM	20% holdout	Response+All Utterances	300	0.673	0.674	0.672
BiLSTM	20% holdout	Response+All Utterances	1213	0.671	0.674	0.669
SVM	20% holdout	Response+All Utterances	1213	0.676	0.676	0.676
Baseline	-	-	-	-	-	0.670

Table 4: Results on Twitter test Dataset

Classifier	Validation Technique	Amount of conversational context used	Max Seq Length	Precision	Recall	F1
BERT	5-fold CV	Only Response	70	**0.658**	**0.658**	**0.658**
BERT	5-fold CV	Response+Last Utterance	120	0.635	0.636	0.635
BERT	5-fold CV	Response+Last 2 Utterances	180	0.491	0.490	0.478
BERT	5-fold CV	Response+Last 3 Utterances	260	0.500	0.250	0.334
BERT	5-fold CV	Response+All Utterances	150	0.595	0.605	0.585
BERT	20% holdout	Response+All Utterances	150	0.587	0.591	0.583
Baseline	-	-	-	-	-	0.600

Table 5: Results on Reddit test Dataset

Twitter data set when only the response and the last utterance in the conversational dialogue was used. The maximum sequence length of 140 was used for this run. This result, however, is close to the F-score of 0.741 obtained using only the response (without any conversational context). On including the last two and the last three utterances from the dialogue, the performance of the BERT classifier degraded considerably with F-score of 0.500 and 0.334 respectively. The reason for this could be that the sequence length was increased for these runs to accommodate the extra contextual information. However, the majority of the instances were of shorter length. Thus, the zero-padding performed on the shorter instances might have degraded the performance. However, it was also found that the performance of the classifier improved considerably (compared to including last two and last three utterances as mentioned above) to an F-score of 0.734 when all the utterances in the context were used and the maximum sequence length was set to 300.

The BiLSTM and SVM classifier obtained F-scores of 0.672 and 0.676 respectively on the Twitter data set. All the utterances were used for these runs.

As can be seen from table 5, the best F-score of 0.658 was obtained for the Reddit data set when only the response was used without any utterance from the dialogue. The maximum sequence length was set to 70 for this run. On using the last utterance along with the response, an F-score of 0.635 was obtained. Just like it happened for the Twitter data set, the performance of the classifier degraded to F-score of 0.478 and 0.334 when the last two and the last three utterances were used respectively. On using all the utterances with a maximum sequence length of 150, the performance again improved to 0.585.

Overall, our best performing runs performed better than the base line scores that were obtained using a BiLSTM with attention based classifier (Ghosh et al., 2018). The classifiers obtained the ranks 14/36 and 21/37 in the leaderboard. However, as our best performing runs were submitted beyond the last date of the competition they have been removed from the leaderboard.

7 Conclusion

In our work, we found that including context in the form of the last utterance in a dialogue chain slightly improved the performance of the BERT classifier for the Twitter data set compared to just using the response alone. For the Reddit data set, including the context did not improve the performance. The best performance for the Reddit data set was obtained when using only the response. Approaches other that only concatenating the utterances to make use of the context needs to be investigated as future work.

References

David Bamman and Noah A. Smith. 2015. Contextualized sarcasm detection on twitter. In *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, pages 574–577. AAAI Press.

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*, 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.

Debanjan Ghosh, Alexander Richard Fabbri, and Smaranda Muresan. 2017. The role of conversation context for sarcasm detection in online interactions. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, pages 186–196. Association for Computational Linguistics.

Sam Glucksberg. 2001. *Understanding Figurative Language: From Metaphor to Idioms*. Oxford University Press, New York.

Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. CASCADE: contextual sarcasm detection in online discussion forums. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1837–1848. Association for Computational Linguistics.

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

Suzana Ilic, Edison Marrese-Taylor, Jorge A. Balazs, and Yutaka Matsuo. 2018. Deep contextualized word representations for detecting sarcasm and irony. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 2–7. Association for Computational Linguistics.

Aditya Joshi, Pushpak Bhattacharyya, and Mark James Carman. 2017. Automatic sarcasm detection: A survey. *ACM Comput. Surv.*, 50(5):73:1–73:22.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan. N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.