Detecting Sarcasm in Conversation Context Using Transformer-Based Models

Adithya Avvaru[1,2], **Sanath Vobilisetty**[2] and **Radhika Mamidi**[1]
[1] International Institute of Information Technology, Hyderabad, India
[2]Teradata India Pvt. Ltd, India
[1]adithya.avvaru@students.iiit.ac.in [1]radhika.mamidi@iiit.ac.in
[2]sanath.vobilisetty@teradata.com

Abstract

Sarcasm detection, regarded as one of the sub-problems of sentiment analysis, is a very typical task because the introduction of sarcastic words can flip the sentiment of the sentence itself. To date, many research works revolve around detecting sarcasm in one single sentence and there is very limited research to detect sarcasm resulting from multiple sentences. Current models used Long Short Term Memory (Hochreiter and Schmidhuber, 1997) (LSTM) variants with or without attention to detect sarcasm in conversations. We showed that the models using state-of-the-art Bidirectional Encoder Representations from Transformers (Devlin et al., 2018) (BERT), to capture syntactic and semantic information across conversation sentences, performed better than the current models. Based on the data analysis, we estimated that the number of sentences in the conversation that can contribute to the sarcasm and the results agrees to this estimation. We also perform a comparative study of our different versions of BERT-based model with other variants of LSTM model and XLNet (Yang et al., 2019) (both using the estimated number of conversation sentences) and find out that BERT-based models outperformed them.

1 Introduction

For many NLP researchers from both academia and industry, sarcasm detection has been one of the most focused areas of research among many research problems like code-mixed sentiment analysis (Lal et al., 2019), detection of offensive or hate speeches (Liu et al., 2019), question-answering(Soares and Parreiras, 2018), etc. One of the main reasons why sarcasm finds a significant portion of research work is because of its nature that the addition of a sarcastic clause or a word can alter the sentiment of the sentence.
Sarcasm is used to criticize people, to provide political or apolitical views, to make fun of ideas, etc., and the most common form of sarcasm usage is through text. Some major sources of the sarcastic text are social media platforms like Twitter, Instagram, Facebook, Quora, WhatsApp etc. Out of these, Twitter forms the major source of sarcastic content drawing attention from researchers across the globe (Bamman and Smith, 2015; Rajadesingan et al., 2015; Davidov et al., 2010).
Due to its inherent nature of flipping the context of the sentence, sarcasm in a sentence is difficult to detect even for humans (Chaudhari and Chandankhede, 2017). Here, the context is considered only in one sentence. How do we deal with situations where the sarcastic sentence depends on a conversation context and the context spans over multiple sentences preceding the response sarcastic sentence? Addressing this problem may help in identifying the root cause of sarcasm in a larger context, which is even tougher because conversation sentences differ in number, some conversation sentences themselves may be sarcastic and response text may depend on more than one conversation sentences. This is the research problem that we are trying to address and are largely successful in building better models which outperformed the baseline F-measures of 0.6 for Reddit and 0.67 for Twitter datasets (Ghosh et al., 2018). We have achieved F-measures of 0.752 for Twitter and 0.621 for Reddit datasets.

2 Related work

Sarcasm is a form of figurative language where the meaning of a sentence does not hold and the interpretation is quite contrary. A quick survey about sarcasm detection and some of the earlier approaches is compiled by Joshi et al. (2017).
The problem of sarcasm detection is targeted in

Proceedings of the Second Workshop on Figurative Language Processing, pages 98–103
July 9, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

Field	Field Description
label	SARCASM or NOT_SARCASM
response	Tweet or a Reddit post
context	Ordered list of dialogue

Table 1: Fields used in the training data

different ways by the research community. Sarcasm detection is not wholly a linguistic problem but extra-lingual features like author and audience information, communication environment etc., also play a significant role in sarcasm identification (Bamman and Smith, 2015). Davoodi and Kosseim (2017) used semi-supervised approaches to detect sarcasm. Another approach is automatic learning and exploiting word embeddings to recognize sarcasm (Amir et al., 2016). Emojis also have a significant impact on the sarcastic nature of the text, which might help in detecting sarcasm better (Felbo et al., 2017). Other approaches to detect sarcasm include Bi-Directional Gated Recurrent Neural Network (Bi-Directional GRNU) (Zhang et al., 2016). Sarcasm detection in speech is also gaining importance (Castro et al., 2019).

Some of the earlier works involving conversation contexts in detecting sarcasm are trying to model conversation contexts and understand what part of conversation sentence was involved in triggering sarcasm (Ghosh et al., 2017, 2018) and identify the specific sentence that is sarcastic given a sarcastic post that contains multiple sentences (Ghosh et al., 2018). Humans could infer sarcasm better with conversation context which emphasises the importance of conversation context (Wallace et al., 2014).

The structure of the paper is as follows. In Section 3, we describe the dataset (fields provided in the train and the test data and an example data along with its explanation). Section 4 describes the feature extraction where the emphasis is on data preprocessing and the procedure to select conversation sentences. Section 5 describes the systems used in training the data whereas section 6 discusses the comparative results of various models. Section 7 presents concluding remarks and future direction of research.

3 Dataset Description

The data[1] we used for model building is taken from sarcasm detection shared task of the Second Workshop on Figurative Language Processing (FigLang2020). There are two types of data provided by the organizers: 1. Twitter dataset and 2. Reddit dataset. Training data contains the fields - "label", "response" and "context" and are described as shown in the Table 1.

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". Consider the example provided by the organizers:

label: *"SARCASM"*

response: *"Did Kelly just call someone else messy? Baaaahaaahahahaha"*

context: [*"X is looking a First Lady should . #classact*, *"didn't think it was tailored enough it looked messy"*]

This example can be understood as *"Did Kelly..."* is a reply to its immediate context *"didn't think it was tailored..."* which is a reply to *"X is looking..."*. and the label of the response is *"SARCASM"*.

Testing data contains the fields - "id", "response" and "context" and are described as shown in the Table 2.

The data of both Twitter tweets and Reddit posts were organized into train and test sets. The number of samples in each of these datasets is shown in Table 3. It is clear from the table that the data is balanced with the same number of sarcastic and non-sarcastic samples (Abercrombie and Hovy, 2016).

Field	Field Description
id	Identification for each test sample
response	Tweet or a Reddit post
context	Ordered list of dialogue

Table 2: Fields used in the testing data

Datasets	Label	No. of Samples	
		Train	Test
Twitter	S	2500	1800
	NS	2500	
Reddit	S	2200	1800
	NS	2200	

Table 3: Dataset Composition Description
∗ *S : SARCASM, NS : NON_SARCASM*

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

4 Feature Extraction

4.1 Data Pre-processing and Cleaning

The corpus data contains consecutive occurrences of periods (.), multiple spaces between words, more or consecutive punctuation marks like exclamation (!), etc. Since the data is collected from Twitter handles and Reddit posts, the data also contain hashtags and emoticons, which are some of the properties of the text extracted from social media. Hence, there is a great need to clean the data before any further processing and we followed multiple steps, for cleaning the data, as described below:

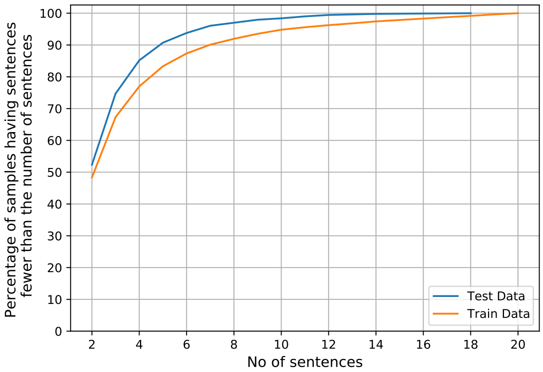

Figure 1: **Analysis of Twitter data:** Number of sentences Vs Percentage of samples

1. Replacing consecutive instances of punctuation marks with only one instance of it.
2. Demojizing the sentences that contain emoticons i.e., replacing emoticons with their corresponding texts. For example, 😜 is replaced with *:stuck_out_tongue:*.
3. There are two ways of handling hashtags - one, remove the hashtag and two, extract the hashtag content. We took the second approach as we believe certain hashtags contain meaningful text. For example, consider the text *Made $174 this month, I'm gonna buy a yacht! #poor*. There are two parts to this sentence - *Made $174 this month*, which doesn't have any sentiment but it is understood that the money he got is less and the second one, *I'm gonna buy a yacht!*, which is a positive statement that he can buy something very costly. The addition of hashtag *#poor* flipped the first statement to negative sentiment. Ignoring *#poor* will lose the sarcastic impact on the sentence.

No of sentences in a conversation	**Training**	**Testing**
5 or less	83%	90%
7 or less	90%	96%
10 or less	95%	98%

Table 4: Twitter Data - Percentage of samples having certain number of sentences in a conversation

No of sentences in a conversation	**Training**	**Testing**
5 or less	99.4%	70%
7 or less	99.9%	93.8%
10 or less	100%	99%

Table 5: Reddit Data - Percentage of samples having certain number of sentences in a conversation

4. Some punctuation marks like exclamation (!) have special significance in English text and are generally used to express emotions such as sudden surprises, praises, excitement or even pain. So, we decided to not remove punctuation marks.
5. We have identified contracted and combined words (for example, *we've*, *won't've*, etc,.) and replaced them with their corresponding English equivalents (in this case, *we have*, *will not have*, etc,.).

4.2 Selection of Conversation Sentences

Twitter Dataset: Since the number of conversation sentences range from two to twenty, it is important to understand how many sentences can contribute to the sarcastic behavior.

A quick analysis of Twitter data is provided by the Figure 1 and the Table 4. The behavior of training and testing data follows similar trend as observed from the Figure 1. We selected the last 7 conversation sentences out of all conversation sentences per Twitter tweet based on the following analysis:

- If we have chosen to select 10 sentences or more, then around 50 percent of samples which have 2 context sentences should be padded with zeros after tokenization. If we have chosen to select 2 sentences, then we will end up losing more context information. There is this trade-off while selecting conversation sentences.

- It is unlikely that the response text depends on the farther context sentences. So, the response text largely depends on context sentences that are closest to the response text.

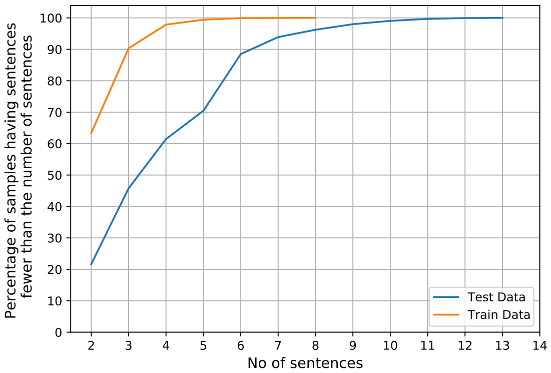

Figure 2: **Analysis of Reddit data:** Number of sentences Vs Percentage of samples

Reddit Dataset: Here, the dataset composition is different compared to that of Twitter Dataset. The number of conversation sentences ranges from two to eight in train data with 99 percent of samples having five or fewer sentences but the number of conversation sentences in test data ranges from two to thirteen with only 70 percent of samples having five or fewer sentences. Figure 2 and the Table 5 depict this behaviour of Reddit data.

4.3 Training text finalization

As discussed in Section 4.2, we considered the last 7 cleaned sentences from the conversation sentences. The response text is a direct result of the conversation sentences. Hence, we concatenate all the selected conversation sentences together and with the cleaned response text. This final text is fed to the model for training.

5 System description

There are several NLP models at our disposal to work with, some are pre-trained while others need to be trained from scratch. We have done experiments with LSTM, BiLSTM, Stacked LSTM and CNN-LSTM (Convolution Neural Network + LSTM) models which can be trained to capture sequence information. To avoid over-fitting, we have introduced dropout layers and taken early stopping measures while training. We split the training data into train data (to train the model) and validation data (10 percent of actual training data to validate the model and employ early stopping). We also have worked with pre-trained Transformer based BERT (bert-base-uncased) model and XLNet. The following steps are used to fine-tune the pre-trained BERT model:

1. Tokenize the text (BERT requires the text to be in a predefined format with separators and class labels)
2. Create attention masks
3. Fine-tune the pre-trained BERT model so that the model parameters will conform to the input training data

In our model, training stops when F1-score on validation data goes below the earlier epoch's F1-score and the prediction is done on the earlier model for which validation F1-score is highest. Similar steps are performed to fine-tune XLNet model.

6 Results

The LSTM model variants - LSTM, BiLSTM, Stacked-LSTM and Conv-LSTM models are applied to Twitter dataset and the F1-scores on test data are 0.67, 0.66, 0.66 and 0.67 respectively. The F1-scores of variants of BERT models considering different lengths of conversation sentences and XLNet are depicted in Table 6.

	Twitter	**Reddit**
BERT-3	0.710	0.603
BERT-5	0.745	**0.621**
BERT-7	**0.752**	-*
BERT-all	0.724	0.592
XLNet	0.684	0.541

Table 6: Comparison of results for various models for Twitter and Reddit datasets
$*$ indicates that the BERT-7 model is not trained as the number of samples in BERT-all model is just one sample more than that in BERT-7 model.

We experimented by considering the last 3, 5 and 8 sentences for Reddit dataset and found that model that used 5 sentences outperformed the other two, probably because the model which used 3 sentences captured the context well while training but failed to apply it as the range of sentences' length in the test set is large compared to the train set. Similarly model with 8 samples had a lot of padded zeros as 99 percent of samples have five or fewer sentences which resulted in poor performance. The results

of the experiments on Reddit dataset are depicted in Table 6. Since LSTM variants did not perform well compared to BERT-based models, we focused more on data preparation part of our research work for Reddit dataset.
It can be inferred from the results table that our hypothesis of taking seven latest sentences, for Twitter dataset, falls in-line with the results.

7 Conclusion

Sarcasm detection in conversational context is an important research area which infuses more enthusiasm and encourages the researchers across the globe. We build models that outperformed the baseline results. Though the results in the Shared Task leaderboard shows that the top model achieved F-measure of 0.93 for the Twitter dataset and 0.83 for the Reddit dataset, there is a lot to work on the problem and find ways to improve the performance with a larger dataset. Use of a larger dataset might help in adding more context and help in improving accuracy. Currently, the models that are built are not generalised across datasets. Further research can focus on building a generalized model for multiple datasets.

References

Gavin Abercrombie and Dirk Hovy. 2016. Putting Sarcasm Detection into Context: The Effects of Class Imbalance and Manual Labelling on Supervised Machine Classification of Twitter Conversations. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 107–113.

Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 167–177, Berlin, Germany. Association for Computational Linguistics.

David Bamman and Noah A Smith. 2015. Contextualized Sarcasm Detection on Twitter. In *Ninth International AAAI Conference on Web and Social Media*.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards Multimodal Sarcasm Detection (An _Obviously_ Perfect Paper). *arXiv preprint arXiv:1906.01815*.

P. Chaudhari and C. Chandankhede. 2017. Literature survey of sarcasm detection. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 2041–2046.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116. Association for Computational Linguistics.

Elnaz Davoodi and Leila Kosseim. 2017. Automatic identification of AltLexes using monolingual parallel corpora. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 195–200, Varna, Bulgaria. INCOMA Ltd.

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*.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

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. *arXiv preprint arXiv:1707.06226*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic Sarcasm Detection: A Survey. *ACM Computing Surveys (CSUR)*, 50(5):1–22.

Yash Kumar Lal, Vaibhav Kumar, Mrinal Dhar, Manish Shrivastava, and Philipp Koehn. 2019. De-mixing sentiment from code-mixed text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 371–377, Florence, Italy. Association for Computational Linguistics.

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm Detection on Twitter: A Behavioral Modeling Approach. In *Proceedings of the eighth ACM international conference on web search and data mining*, pages 97–106.

Marco Antonio Calijorne Soares and Fernando Silva Parreiras. 2018. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University-Computer and Information Sciences*.

Byron C Wallace, Laura Kertz, Eugene Charniak, et al. 2014. Humans require context to infer ironic intent (so computers probably do, too). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 512–516.

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.

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.