

# C-Net: Contextual Network for Sarcasm Detection

Amit Kumar Jena, Aman Sinha\* and Rohit Agarwal\*

Indian Institute of Technology (Indian School of Mines) Dhanbad, India  
{amitjena40, amansinha091, agarwal.102497}@gmail.com

## Abstract

Automatic Sarcasm Detection in conversations is a difficult and tricky task. Classifying an utterance as sarcastic or not in isolation can be futile since most of the time the sarcastic nature of a sentence heavily relies on its context. This paper presents our proposed model, C-Net, which takes contextual information of a sentence in a sequential manner to classify it as sarcastic or non-sarcastic. Our model showcases competitive performance in the Sarcasm Detection shared task organised on CodaLab and achieved 75.0% F1-score on the Twitter dataset and 66.3% F1-score on Reddit dataset.

## 1 Introduction

Sarcasm detection plays a crucial role in improving the effectiveness of chatbot systems. Sentiment classification systems can fail in the absence of a robust sarcasm detection system. A sarcastic sentence can express a negative sentiment even with the presence of positive or neutral sentiment words in that sentence. Hence, accurate detection of sarcasm can take an artificially intelligent agent closer to imitate human behaviour and enable it to better understand true intentions and emotions of humans (Joshi et al., 2018).

This paper represents work on the Sarcasm Detection shared task which is a part of the Second Workshop on Figurative Language Processing, co-located with ACL 2020. The shared task aims to investigate and understand how much conversation context is needed or helpful for Sarcasm Detection. Two datasets, one of Reddit and the other of Twitter, were provided for developing and testing multiple sarcasm detection systems.

In this paper, we present our study on the effectiveness of contextual information to decide if an

utterance is sarcastic or not. For this, the baseline models were first created using traditional machine learning algorithms like logistic regression, SVM etc. which were trained to classify utterances without considering their contextual information. Sequence models like vanilla RNN and LSTM were trained similarly. Then different types of word embeddings (ELMo and Glove) and sentence embedding (DeepMoji) to capture emotional states in the sentences were also experimented to detect incongruities within the text. The latest state-of-the-art transformer based models like BERT, XLNet and RoBERTa were also used for classifying sentences in isolation. Our investigations for creating systems which can use the context information effectively in a sequential manner led to the creation of our proposed model, which showed decent performances in both the test datasets.

## 2 Related Works

The evolution of various trends in Sarcasm detection research can be seen in (Joshi et al., 2017). Sarcasm detection was initially performed using rule-based approaches. (Riloff et al., 2013) presented rule-based classifiers that look for a positive verb and a negative situation phrase in a sentence. (Maynard and Greenwood, 2014) proposed using hashtag sentiment as an indicator for sarcasm and (Liu et al., 2014) introduced POS sequences and semantic imbalance as features. Statistical feature-based approaches were also used for this task e.g. (Reyes and Rosso, 2012) introduced features related to ambiguity, unexpectedness, emotional scenario, etc. to better capture situational dependencies for the presence of sarcasm.

Machine learning algorithms were also used for sarcasm detection. Majority of work in sarcasm detection earlier relied on SVM ((Joshi et al., 2015); (Tepperman et al., 2006); (Kreuz and Caucci,

\* Equal contribution.

Dataset	Train	Test
Reddit	4400	1800
Twitter	5000	1800

Table 1: Dataset statistics.

2007); (Tsur et al., 2010); (Davidov et al., 2010)). (Riloff et al., 2013) compared rule-based techniques with an SVM-based classifier. (Reyes et al., 2013) used Naive Bayes and decision trees for multiple pairs of labels among irony, humor, politics and education. (Bamman and Smith, 2015) used binary logistic regression for their work.

The importance of context information was first presented in (Wallace et al., 2014) which described their annotation study where annotators repeatedly asked for context information to judge a text to be sarcastic or not. Many times they changed their previously given labels to a text after being shown the context behind it. (Rajadesingan et al., 2015) and (Bamman and Smith, 2015) tried to include the author context by analysing the author’s past tweets and sentiments. To consider the conversational context, (Wang et al., 2015) and (Joshi et al., 2016) used a sequence labeling approach. (Wang et al., 2015) also tried to use the topical context of a text, since some topics are more likely to generate sarcasm as compared to other topics.

Recent works in this domain include deep learning methods such as (Ghosh et al., 2018) included several types of Long Short-Term Memory (LSTM) networks that can model both, the conversation context and the response. (Hazarika et al., 2018) used CNNs to incorporate various contextual information. (Potamias et al., 2019) used pre-trained RoBERTa weights combined with an RCNN to capture contextual information to detect sarcasm.

### 3 Data

The dataset used for this study was provided in the shared task on Sarcasm Detection, organized at Codalab. It included two separate datasets, Twitter and Reddit, each of them having equal no. of sarcastic and non-sarcastic responses. For each response provided in the dataset, the conversation context consists of an ordered list of previous dialogues. Table 1 shows the size of the train and test sets of both the datasets. From Figure 1, we can see that we have variable number of sentences in the context set of responses, ranging from 2 to 20.

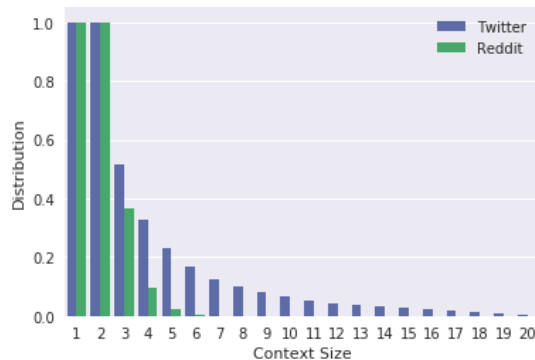


Figure 1: Context set size distribution. The x-axis shows the size of context sets in both the training datasets. The y-axis shows the percentage of data containing that much context size.

## 4 Methods

We used three kinds of approaches to experiment in this task. First, methods that classified utterances in isolation were investigated. Then, approaches that considered partial conversation context for classifying texts were experimented. Finally, methods that can potentially utilise the complete conversation context information were looked into.

### 4.1 Baseline models

We experimented with traditional machine learning based approaches like logistic regression, Naive Bayes classifier, SVM etc. first for sarcasm detection by treating the response sentences in isolation. Sequential models like RNNs can easily model a sequential data hence, they are widely used in Natural Language Processing. Basic RNN and LSTM variants were also used in experiments for this task.

### 4.2 Pretrained Networks

We used Deepmoji (Felbo et al., 2017) in order to investigate the correlation between emotion and presence of sarcasm in sentences. ELMo (Peters et al., 2018) provides contextualized word representation where embeddings of each word is actually a function of the entire sentence containing that word. This may help in capturing local semantic incongruities within a sentence, which is an indicator of sarcasm. Recently introduced transformer models like BERT (Devlin et al., 2019), XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019) have given state-of-the-art results on various NLP tasks. Experiments were performed with these models to classify utterances as sarcastic or not-sarcastic.

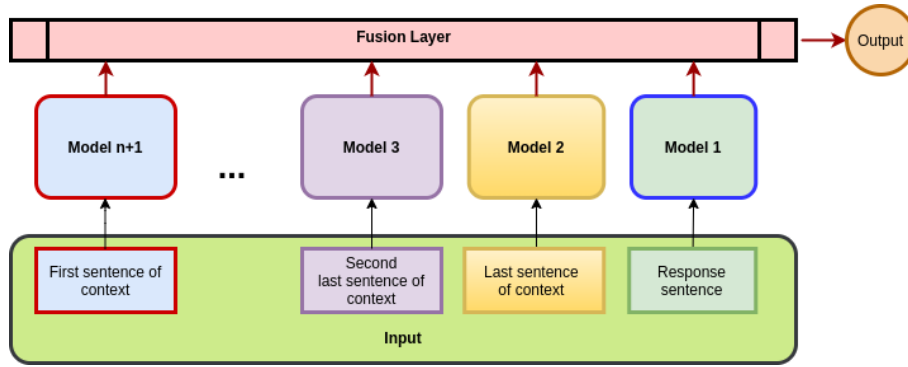


Figure 2: **C-Net Architecture:** Here, ‘n’ is the maximum size of the context set. Model-1, 2, 3... n+1 are BERT (base-uncased) models which are trained separately on the response sentences, last sentence of context sets, second last sentence of context sets and so on till the first sentence of context sets respectively. Probability values generated by these n+1 models are used by the Fusion Layer to generate another probability value as output, which tells about the possibility of sarcasm presence in the response.

### 4.3 C-Net

Sarcasm detection can be attributed to information like emotional state, the topic of the conversation, etc. which can be extracted from the conversation context of an utterance. Manually annotating a huge corpus of text data can be a tedious task. We propose our model Contextual-Network (C-Net) for Sarcasm Detection, which uses pseudo-labeling to provide labels for the context sentences by giving them the same label as the response sentence. This is followed by a fusion layer as shown in Figure 2. Training in this way helps in including the contextual information in the model and therefore aid in detecting situations that may lead to the occurrence of a sarcastic sentence in the near future.

By using pseudo labels for training BERT on context sentences, we assigned a score to each context sentence. These scores told about the probability of the conversation leading to a sarcastic response, if that particular context sentence was present in the conversation. This helped in analysing how sarcasm generating situations build up during conversations.

For this model, we used Simple Exponential Smoothing (SES) in the fusion layer, which is a time series forecasting method for univariate data without a trend or seasonality. Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older.

The mathematical expression for Simple Exponential Smoothing (SES) is given by

$$y_{t+1} = \alpha(y_t + (1 - \alpha)y_{t-1} + (1 - \alpha)^2 y_{t-2} + \dots)$$

where  $\alpha \in (0, 1)$  controls the rate at which the

influence of the observations at previous time steps decays exponentially. Here  $y_{t-1}$ ,  $y_{t-2}$ , and so on, are scores predicted by Model 2, Model 3 and so on till Model n+1 respectively. These scores are the probability of the response being sarcastic if these context sentences were present in the conversation anytime before the response.  $y_t$  is the probability of response being sarcastic, predicted by Model 1. C-Net takes all these scores into consideration and gives the final output value  $y_{t+1}$ , as shown in Figure 2. In this way the method is capable of handling the complete context set.

Hence, generating probability values by giving pseudo-labels to context sentences and combining those values using simple exponential smoothing helps in making a more accurate prediction of sarcasm in conversation.

## 5 Implementation

### 5.1 Methods using response only

The training datasets were split into a 90% training set and 10% validation set. We used Fastai tokenizer for pre-processing the datasets and then applied various basic machine learning algorithms for sentence classification. We also used the torch-text library and spacy tokenizer to pre-process the dataset before using Vanilla RNN and bidirectional LSTM models. Both the models were trained for 10 epochs.

We used transformer architectures available within the Huggingface’s transformers library and trained them using Gradual Unfreezing and Slanted Triangular Learning Rates (Howard and Ruder, 2018). The learning rate for the last layer was

Method	Twitter	Reddit
<i>Response Only Set</i>		
Logistic Regression	0.685	0.622
Naive Bayes	0.673	0.626
SGD Classifier	0.668	0.626
XGBoost	0.672	0.617
SVM	0.632	0.334
Vanilla RNN	0.478	0.463
Bi-LSTM	0.497	0.481
DeepMoji	0.679	0.633
ELMo	0.684	0.544
ELMo+DeepMoji	0.681	0.518
XLNet (base-cased)	0.712	0.598
BERT (base-uncased)	<b>0.733</b>	0.671
RoBERTa (base)	0.680	<b>0.678</b>
<i>Fixed Context Set</i>		
C-Net+LR	0.747	0.650
C-Net+SES	<b>0.750</b>	<b>0.663</b>
<i>Complete Context Set</i>		
Time-stamping	0.710	0.500

Table 2: Results on test datasets (F1-scores)

1e-4 and for other layers, it was 1e-5. The batch size used was 16 and the model was trained with half-precision settings on 16 GB GPU.

Pre-trained torchMoji was used to generate 2304 dimensional sentence encoding for each response sentence. Using ELMo we obtained 1024 dimensional word vectors. The sentence vectors for each response was obtained by averaging the word-vectors. We also concatenated the ELMo representation of each response sentence with the DeepMoji representation of the same sentence to make the sentence representations richer. Then we applied logistic regression to classify the sentence representations obtained by the above-said approaches.

## 5.2 Methods using fixed context set

We observe that two fixed context utterances are always available for each response in both the datasets. Thus, we create a C-Net with 3 models. Model 1 uses response, Model 2 uses the latest context and Model 3 uses the second latest context. For the training of each model, the output target is the label of response. Thus, we train each model in the lieu of detecting sarcasm in response.

As we see in Figure 2, the fusion layer works on the probability values generated by the BERT (base-uncased) models to give an output. The fusion layer can be either a Logistic Regression or a Simple

Exponential Smoothing model. Since the sequence of dialogues in a conversation matters in deciding the polarity or emotion of future dialogues, simple exponential smoothing was used to take advantage of the sequential nature of the dataset.

## 5.3 Method using complete context set

In order to include the complete context set for training, we used Timestamping to preserve the sequence of sentences. In this method, two bert-base-uncased models were trained separately on the response only set and all the context sentences. Also, for all the context sentences, a special marker was concatenated at the end which would make the model aware of the position of that sentence in a conversation. Output probabilities by these two models were used to get label for the response.

## 6 Reproducibility

To reproduce the best results mentioned in this paper, C-Net with SES should be used, utilising the two latest context sentences associated with each response. The pre-trained BERT (base-uncased) model, provided in the transformers library by Huggingface (Wolf et al., 2019), was trained similarly as mentioned in the implementation section using the fastai library (Howard and Gugger, 2020). While fine-tuning BERT on response sentences, the learning rates used were ranging between 1e-5 to 1e-4. But while training on Context sentences, the learning rates used were ranging between 1e-6 to 1e-5. The optimum parameter  $\alpha$  for SES was found out from the training set with a grid search. In our experiments, it was found out that the value 0.395 for  $\alpha$  best fits the Twitter training data and 0.2 best fits the Reddit training data.

## 7 Results

From Table 2, it can be seen that the BERT classifier performed the best on Twitter dataset and RoBERTa model performed the best on the Reddit dataset when compared to all methods over *response only* set. Overall, the C-Net model gave the best results as compared to all the approaches on the twitter test dataset.

The results in the case of C-Net for both the fusion methods (LR and SES) are better as compared to the results of the BERT classifier trained only on response in the twitter data. However, this is not true for the Reddit dataset. The BERT

and RoBERTa model trained only on response sentences in the Reddit dataset performed better as compared to the C-Net approach. This is counter-intuitive as per the theory that context information helps in sarcasm detection. However, it's possible that the Reddit response-only dataset contains many flags for sarcasm, which are also present in the large dataset the models were pre-trained with. Further pre-training of the models on the target Reddit and Twitter dataset may further improve the results.

## 8 Conclusion

In this paper, we compared the performances of various approaches for the Sarcasm Detection task. We experimented with traditional machine learning based approaches, and the latest state-of-the-art transformer architectures. The results obtained show that our proposed model, C-Net, has the potential to effectively use the conversation context of an utterance to capture the sarcastic nature of a conversation.

The variable number of context sentences for each response sentence makes it difficult to capture the long range dependency. Hence, as future work, approaches that can effectively deal with variable context set size can be investigated.

## References

- David Bamman and Noah A. Smith. 2015. Contextualized sarcasm detection on twitter. In *ICWSM*.
- 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, CoNLL '10*, page 107–116, USA. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/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](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.
- Debanjan Ghosh, Alexander R. Fabbri, and Smaranda Muresan. 2018. [Sarcasm analysis using conversation context](#). *Computational Linguistics*, 44(4):755–792.
- Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. [Cascade: Contextual sarcasm detection in online discussion forums](#). *ArXiv*, abs/1805.06413.
- Jeremy Howard and Sylvain Gugger. 2020. fastai: A layered api for deep learning. *Information*, 11:108.
- Jeremy Howard and Sebastian Ruder. 2018. [Fine-tuned language models for text classification](#). *ArXiv*, abs/1801.06146.
- Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. [Automatic sarcasm detection: A survey](#). *ACM Comput. Surv.*, 50(5).
- Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2018. *Investigations in Computational Sarcasm*, 1st edition. Springer Publishing Company, Incorporated.
- Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. [Harnessing context incongruity for sarcasm detection](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, Beijing, China. Association for Computational Linguistics.
- 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.
- Roger J. Kreuz and Gina M. Caucchi. 2007. Lexical influences on the perception of sarcasm. In *Proceedings of the Workshop on Computational Approaches to Figurative Language, FigLanguages '07*, page 1–4, USA. Association for Computational Linguistics.
- Peng Liu, Wei Chen, Gaoyan Ou, Tengjiao Wang, Dongqing Yang, and Kai Lei. 2014. Sarcasm detection in social media based on imbalanced classification. In *Web-Age Information Management*, pages 459–471, Cham. Springer International Publishing.
- 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. *ArXiv*, abs/1907.11692.
- Diana Maynard and Mark Greenwood. 2014. [Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4238–4243, Reykjavik, Iceland. European Language Resources Association (ELRA).

- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Rolandos Alexandros Potamias, Georgios Siolas, and A. Stafylopatis. 2019. [A transformer-based approach to irony and sarcasm detection](#). *ArXiv*, abs/1911.10401.
- 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, WSDM '15*, page 97–106, New York, NY, USA. Association for Computing Machinery.
- Antonio Reyes and Paolo Rosso. 2012. [Making objective decisions from subjective data: Detecting irony in customer reviews](#). *Decision Support Systems*, 53:754–760.
- Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. [A multidimensional approach for detecting irony in twitter](#). *Lang. Resour. Eval.*, 47(1):239–268.
- Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. [Sarcasm as contrast between a positive sentiment and negative situation](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.
- Joseph Tepperman, David Traum, and Shrikanth Narayanan. 2006. Yeah right: Sarcasm recognition for spoken dialogue systems.
- Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. [Icwsn - a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews](#). In *ICWSM*.
- Byron C. Wallace, Do Kook Choe, Laura Kertz, and Eugene Charniak. 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, Baltimore, Maryland. Association for Computational Linguistics.
- Zelin Wang, Zhijian Wu, Ruimin Wang, and Yafeng Ren. 2015. [Twitter sarcasm detection exploiting a context-based model](#). In *Proceedings, Part I, of the 16th International Conference on Web Information Systems Engineering — WISE 2015 - Volume 9418*, page 77–91, Berlin, Heidelberg. Springer-Verlag.
- 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.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *NeurIPS*.