A Novel Hierarchical BERT Architecture for Sarcasm Detection

Himani Srivastava
TCS Research
New Delhi
India
`srivastava.himani`
@tcs.com

Vaibhav Varshney
TCS Research
New Delhi
India
`varshney.v`
@tcs.com

Surabhi Kumari
TCS Research
New Delhi
India
`surabhi.kumari6`
@tcs.com

Saurabh Srivastava
TCS Research
New Delhi
India
`sriv.saurabh`
@tcs.com

Abstract

Online discussion platforms are often flooded with opinions from users across the world on a variety of topics. Many such posts, comments, or utterances are often sarcastic in nature, i.e., the actual intent is hidden in the sentence and is different from its literal meaning, making the detection of such utterances challenging without additional context. In this paper, we propose a novel deep learning-based approach to detect whether an utterance is sarcastic or non-sarcastic by utilizing the given contexts in a hierarchical manner. We have used datasets from two online discussion platform - Twitter and Reddit[1] for our experiments. Experimental and error analysis shows that the hierarchical models can make full use of history to obtain a better representation of contexts and thus, in turn, can outperform their sequential counterparts.

1 Introduction

In the current scenario, social media serves as the biggest platform for people to express their opinion and share information. Many organizations use this data to understand the choices of people and amend their policies accordingly. On these platforms, people often express their opinion sarcastically which is inherently difficult even for humans to analyze. For example *"It is a wonderful feeling to carry an expensive phone with short battery life."* is a sarcastic sentence that complains about the battery life of the phone but with the positive set of words like "wonderful". Therefore it is essential to identify sarcastic responses to comprehend the users' demands and complaints.

However, detecting sarcasm from a text is a difficult task as such sentences have positive surface sentiment but negative implied sentiment. For example, in the sentence *"Yeah Right! I bought that nice expensive phone for this only!"*, the phrase "nice expensive" may imply positive sentiment from the user, but, the phrase "Yeah Right!" may render the whole sentence as a negative statement given enough background or context.

Sarcasm detection is not an independent area of study and is closely related to sentiment analysis. In order to detect sarcasm, many works like, (Veale and Hao, 2010), (Maynard and Greenwood, 2014) have proposed the use of hand-crafted features to identify a sarcastic response.

However, with the advent of Deep Learning, it became possible to automatically learn and extract these features, thereby reducing both time and effort. Many of these approaches have also been applied in complex NLP problems, for example, (Kim, 2014) proposed a Convolution Neural Network (CNN) to extract n-gram features automatically from the text. (Nowak et al., 2017) and (Cho et al., 2014) showed the efficacy of Recurrent Neural Networks (RNNs) like LSTMs (Hochreiter and Schmidhuber, 1997) and GRUs (Cho et al., 2014) in handling the long term dependencies. Attention mechanisms (Bahdanau et al., 2014) have further improved the performance of complex NLP tasks like machine translation and reading comprehension by attending or focusing on the important words/ phrases from the inputs before making a decision. Recently, transformers (Vaswani et al., 2017) have outperformed many traditional and recent approaches in NLP by allowing one to learn from the huge amount of data.

In this paper, we present a Hierarchical BERT (Devlin et al., 2018) based model for sarcasm detection for a given response and its context. Our model, first, extracts the local features from the words in a sentence, and then uses a Convolution module to summarize all the sentences in a context.

[1] Both the dataset are provided by the organizers of Shared Task of Sarcasm Detection in FigLang-2020

Proceedings of the Second Workshop on Figurative Language Processing, pages 93–97
July 9, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

The summarized context is then passed through a recurrent layer to extract the temporal features from the input. These temporal features are then convoluted with the input response to detect whether the response is sarcastic or not.

2 Related Work

The task of sarcasm detection can be formulated as a binary classification task i.e. given a sentence, the task is to predict whether it is sarcastic or not. Another area of study involves labeling utterances in a dialogue as sarcastic or non-sarcastic using sequence labeling. These approaches usually fall into three different categories namely, Rule-based, Statistical, and deep learning-based.

Rule Based Approaches In (Veale and Hao, 2010), (Maynard and Greenwood, 2014), (Bharti et al., 2015) and (Riloff et al., 2013) authors proposed the use of hand-crafted features and rule-based approaches to perform classification. In order to learn a decision boundary, one has to model all the hand-crafted features beforehand, which is a big disadvantage with such approaches.

Statistical Approaches In statistical approaches, features like bag-of-words, pattern-based, user mentions, emoticons, N-grams have been proposed in (Tsur et al., 2010), (González-Ibáñez et al., 2011), (Liebrecht et al., 2013), (Reyes et al., 2013). (Barbieri et al., 2014) included several sets of features such as minimum/ maximum/ average number of synset and synonyms, minimum/ maximum gap of the intensity of adverb and adjectives in the target text to build first automated system targeted for detecting irony [2] in Italian Tweets.

Deep Learning-based Approaches The above mentioned approaches suffer from generalization since it is hard to manually extract and define all the rules and features to detect sarcasm, whereas, deep learning approaches can generalize well by automatically learning from data. (Joshi et al., 2016) used similarity between word embeddings of utterances for sarcasm detection. (Amir et al., 2016) applied convolution operation on user embedding and the utterance embedding for sarcasm detection. User embedding allowed them to learn user-specific context. and auxiliary features [3] to train the convNet. (Cai et al., 2019) used a multi-modal fusion model to detect sarcasm in a tweet that may contain an image or video along with the text. In (Potamias et al., 2019), authors proposed use of transformer for detecting sarcastic text.

3 Dataset

For Sarcasm Detection, we have used 2 datasets namely: (1) Twitter Dataset and (2) Reddit Dataset provided for the FIG-LANG shared task[4]. Both the datasets have a 'Context' provided in the form of a conversation between the users, and the final 'Response' that has to be classified as Sarcastic or Non Sarcastic response, using context. The Twitter and Reddit dataset contains 5000 and 4400 train instances respectively and 1800 test instances each. The train set was further divided into an 80:20 ratio in a stratified fashion to obtain our final dev (evaluation) and train sets.

4 Approach

In our proposed approach, we hypothesize that the context must have a significant role in deciding the sarcastic orientation of the response. Hence, in order to capture the temporal features from the context, we processed the contexts in a hierarchical manner. In this section, we describe all the components of our proposed architecture.

Sentence Encoding Layer: To obtain the initial representation of the input, we used 2 separate encoder layers for the context and its response. The utterances in a context are passed through the first BERT layer to extract sentence level features of a context. For instance, if our context contains 'm' different utterances then, the output of this layer would be $\mathbf{s_{con}} \in R^{(m,d_{sen},d_{bert})}$ where, d_{sen} is the maximum sentence length and d_{bert} is the length of word vector obtained from BERT. In our experiments, we have used *all the context* provided in the input to obtain initial context representation.
Similarly, the second BERT layer is used to encode the response in a fixed length vector $\mathbf{s_{res}} \in R^{d_{sen},d_{bert}}$. This representation is further passed through a BiLSTM layer to capture the semantic relationship between the words of a response. The final response output is denoted by $\mathbf{o_{res}} \in R^{d_{sen},d_{lstm}}$ where, d_{lstm} is the number of BiLSTM units.

Context Summarization Layer: The size of initial context vector $\mathbf{s_{con}}$ after the sentence en-

[2] Sarcasm is a form of irony.

[3] In this paper 5 auxiliary features are taken which are: count of (!, ?, ., capital letters, "or") in a tweet

[4] The shared task on sarcasm detection conducted at the ACL 2020 workshop on Figurative Language Processing.

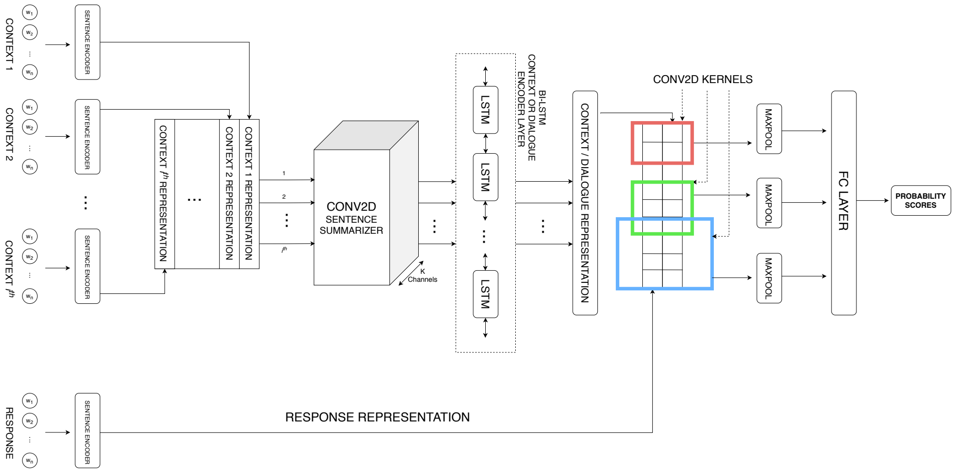


Figure 1: Proposed Architecture

coder layer becomes too large to process. For instance, if d_{bert} is 768, d_{sen} is 100 and m is 10 then our initial representation will be of size 10 × 100 × 768. Thus, in order to obtain a summarized context, we projected the utterances to a lower dimension space using a convolution operation. To achieve this, we passed all the utterances through a 2D convolution layer with kernel size of (k_{row}, k_{col}) and a stride of 1. We obtain d_{sum} such feature maps to output our summary, $sum_{con} \in R^{(m-k_{row}, d_{sen}-k_{col}, d_{sum})}$.

Context Encoder Layer: Since the utterances are sequential in nature, i.e., one utterance is uttered in response to the previous one, it is essential to capture the contextual information between the utterances to obtain a better context representation. We have used BiLSTMs to output a sequence of hidden state vectors $[\mathbf{h^1}, \mathbf{h^2}, ..., \mathbf{h^M}]$ corresponding to each of the M input vectors, where M is $m - k_{row}$. The vector h^M can be seen as a short summary for whole the context just like a short summary of a paragraph or a book. The final output of this layer is $\mathbf{o_{con}} \in R^{(M, d_{lstm})}$

CNN Layer: In (Kim, 2014) author proposed a hybrid multi-channel CNN to capture the N-grams features in a text by varying the kernel size. As our final output $\mathbf{o}$, we again used a 2D convolution layer to extract relations between response $\mathbf{o_{res}}$ and context $\mathbf{o_{con}}$. To obtain different N-grams features as mentioned in (Kim, 2014), we used different shared matrices of sizes (2, 2), (2, 3), and (2, 5). Finally, we applied a Max-pool layer to extract the most relevant features from each of these N-gram features. Our final architecture is described in Figure 1.

Fully Connected Layer: The relevant N-grams from the last layer are then passed through a fully connected layer to obtain a score S, which is then passed through a sigmoid layer to compute the probability scores.

5 Experiments and Results

For our experiments, we compare proposed approach with two baselines defined next:

Baseline 1: Hierarchical Attention Network Proposed in (Yang et al., 2016), the authors applied attention mechanism to classify large documents. We used this model to visualize the attention given to a particular context and words in response while making a decision.

Baseline 2: Memory Networks This experiment is done based on the implementation of (Sukhbaatar et al., 2015). The intuition was to use (context, response) as a (key, value) pair, and given this information we predicted whether our value is sarcastic or not. Results have been shown in the table 1.

Experimental Settings All the parameters in our architecture were tuned on val set. We have used small version of BERT with $d_{bert} = 768$, the number of Bi-LSTMs, d_{lstm} were varied between the range 200, 300. Throughout our experiments on test and val sets, we found that the optimum value for (k_{row}, k_{col}) were (2,2) and were kept same throughout all other experiments. The maximum sentence length, d_{sen} was fixed to 100 and the dropout values were adjusted as described in (Srivastava et al., 2014). We used cross-entropy as the loss function and Adam as optimizer (with default values) for all the models. The F1 scores were used as an evaluation metric for validation set.

All these parameters were tuned on val F1 scores to determine the final optimal values.

6 Results and Analysis

We have reported the results of all the experiments table 1. As shown in table, our architecture outperformed all the other strong baselines in both the datasets. To further analyse the strengths of our network we further performed some experiments by visualizing the attention weights given to contexts and words in response.

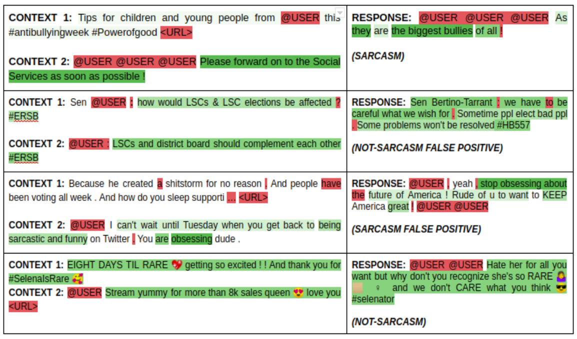
As evidenced from Figure 2, we can see that the model correctly predicts the inputs in row 1 and 4. In row 1, the maximum attention is given to Utterance 2, while in the response, a strong negative phrase like "biggest bullies" were given maximum attention to classify the response as sarcastic. Also, we can see that the irrelevant words like "@USER" were given the least attention (shades of green determine the positive while, the shades of red determine the negative attention weights). Similarly in row 4, we can see that the context and response are consists of positive sentiment words and emoticons which helped the model to classify the input as non-sarcastic.

In row 2 and 3 there are examples of incorrect classification made by our model. Upon analysis, we found that the response contains the positive words used in negative sentiment thereby confusing the model. Similarly in row 3, the response contains the negative words/ phrases like "stop obsessing", "rude" but has positive sentiment. Such examples where, there is a very fine distinction between the sarcastic and non-sarcastic responses are likely to confuse our model.

We hypothesised that the false-positive produced by our models must be are of "boundary cases" which our model is not able to handle. To corroborate this fact, we plotted the embedding produced by our model before sigmoid layer using t-SNE (van der Maaten and Hinton, 2008). In Figure 3, green and blue points denotes correct non-sarcastic and sarcastic samples respectively while orange and red points denote the incorrect classifications. We can see that most of the miss-classifications form a cluster and can be seen as boundary examples. Also, to explain the difference between the val and test results we plotted the test samples on top of these points (denoted by cross markers). We can see in Figure 3, that the test samples fall on these boundary cases which might be the reason for such discrepancy.

Dataset		HAN	KG-Mem	Our Approach
Twitter	train	0.76	0.79	**0.87**
	dev	0.74	0.79	**0.84**
	test	0.68	0.70	**0.74**
Reddit	train	0.69	0.69	**0.77**
	dev	0.67	0.68	**0.76**
	test	0.60	0.605	**0.639**

Table 1: Comparison with baselines



CONTEXT 1: Tips for children and young people from @USER this #antibullyingweek #Powerofgood <URL> **CONTEXT 2:** @USER @USER @USER Please forward on to the Social Services as soon as possible !	**RESPONSE:** @USER @USER @USER As they are the biggest bullies of all ! ***(SARCASM)***
CONTEXT 1: Sen @USER : how would LSCs & LSC elections be affected ? #ERSB **CONTEXT 2:** @USER : LSCs and district board should complement each other #ERSB	**RESPONSE:** Sen Bertino-Tarrant : we have to be careful what we wish for . Sometime ppl elect bad ppl . Some problems won't be resolved #HB557 ***(NOT-SARCASM FALSE POSITIVE)***
CONTEXT 1: Because he created a shitstorm for no reason . And people have been voting all week . And how do you sleep supporti ... <URL> **CONTEXT 2:** @USER I can't wait until Tuesday when you get back to being sarcastic and funny on Twitter . You are obsessing dude .	**RESPONSE:** @USER , yeah . stop obsessing about the future of America ! Rude of u to want to KEEP America great ! @USER @USER ***(SARCASM FALSE POSITIVE)***
CONTEXT 1: EIGHT DAYS TIL RARE ❤️‍🩹 getting so excited ! ! And thank you for #SelenaIsRare 🥰 **CONTEXT 2:** @USER Stream yummy for more than 8k sales queen 😍 love you <URL>	**RESPONSE:** @USER @USER Hate her for all you want but why don't you recognize she's so RARE 🤷 ♀ and we don't CARE what you think 😎 #selenator ***(NOT-SARCASM)***

Figure 2: Error Analysis

7 Conclusion and Future work

In this paper, we proposed a novel Hierarchical BERT based neural network architecture to handle context and response. From analysis and results, we supported the facts that the hierarchical model can effectively model the context and can produce a better representation of input before making a decision.

Applying BERT on large documents and in hierarchical setting is still an open problem and we would like to explore this aspect in depth in our future works. Further, we would like to obtain a better representation of context by compressing the BERT representation of context in a much more efficient way (the context summarization layer).

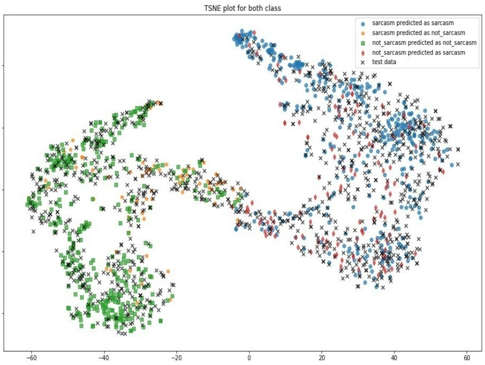


Figure 3: t-SNE plot for val and test data

References

Silvio Amir, Byron Wallace, Hao Lyu, Paula Carvalho, and Mário Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. pages 167–177.

Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. 2014. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409.

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2014. Italian irony detection in twitter: a first approach.

S. K. Bharti, K. S. Babu, and S. K. 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.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multimodal sarcasm detection in twitter with hierarchical fusion model. pages 2506–2515.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 581–586, Portland, Oregon, USA. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Aditya Joshi, Kevin Patel, Vaibhav Tripathi, Pushpak Bhattacharyya, and Mark Carman. 2016. Are word embedding-based features useful for sarcasm detection?

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Christine Liebrecht, Florian Kunneman, and Antal van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 29–37, Atlanta, Georgia. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

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).

Jakub Nowak, Ahmet Taspinar, and Rafal Scherer. 2017. Lstm recurrent neural networks for short text and sentiment classification. pages 553–562.

Rolandos Potamias, Georgios Siolas, and Andreas Stafylopatis. 2019. A transformer-based approach to irony and sarcasm detection.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47.

Ellen Riloff, A. Qadir, P. Surve, L. Silva, N. Gilbert, and R. Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. *Proceedings of EMNLP*, pages 704–714.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm - a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Tony Veale and Yanfen Hao. 2010. Detecting ironic intent in creative comparisons. pages 765–770.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.