Sarcasm Detection (Nitro NLP 2024)

Florin-Silviu Dinu
florin-silviu.dinu@s.unibuc.ro

Alexandru Tindeche
alexandru.tindeche@s.unibuc.ro

Șincarenco Nichita
nichita.sincarenco@s.unibuc.ro

Florin Petrișor Tănasă
florin-petrisor.tanasa@s.unibuc.ro

Abstract

Sarcasm detection is a hard task when it comes to NLP. This paper uses a reference model with BERT, CNN and an attention mechanism and modifies it by making two other models: one only with a CNN but with two convolutional layers and another one with a two convolutional layers CNN and the referenced attention mechanism. It uses a limited dataset and tests if by excluding the attention mechanism from the reference paper better results will be achieved.

1 Introduction

Sarcasm is usually used for mocking someone in order to hurt or amuse that person (Majumdar et al., 2022). The best way to detect sarcasm is through verbal communication because in writing detection becomes more complex (Anup et al., 2020). It's true that many times on social networks discourse, sarcasm can be represented by writing in all caps, using distinctive signs including emojis (Majumdar et al., 2022). This helps overcome the barrier that strips written language of sarcastic meaning but it is not always present.

When it comes to types of sarcasm, 3 categories can be identified (Moores and Mago, 2022)

1. A positive superficial sentiment expressing a negative implicit sentiment (Moores and Mago, 2022)

2. A negative superficial sentiment expressing a positive implicit sentiment (Moores and Mago, 2022)

3. A neutral superficial sentiment expressing a negative implicit sentiment (Moores and Mago, 2022)

According to Rogoz et al. sarcasm is also divided into 3 categories: funny, ridiculous and ironical (Rogoz et al., 2021).

According to Watson there is a clear distinction between satire, sarcasm and irony (Watson, 2011)

Having three distinct classification in three different papers, the question of no standard definition of sarcasm arises. We will proceed to train our models on a labeled dataset but the labeling is therefore done subjectively and all results should be interpreted as such.

2 Related Work

When it comes to sarcasm detection, Moores and Mago presented the following survey Figure 1

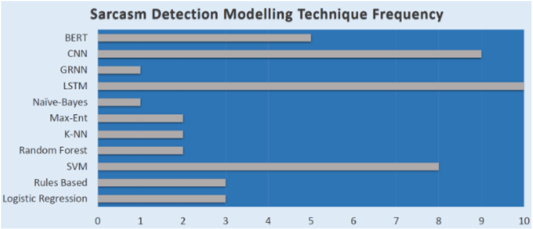

Figure 1: Popularity of sarcasm detection models (Moores and Mago, 2022)

From 1 we find out that LSTM dominates the list as being the most popular, followed by CNN, SVM and BERT.

The name of BERT comes from Bidirectional Encoder Representations from Transformers and has the ability to combine the left and right context of the analysed word in order to train a bidirectional transformer model (Devlin et al., 2019)

One of the BERT models trained on Romanian vocabulary is the one presented by Dumitrescu Ștefan et al. This model is based on Multilingual BERT and has been train on a vocabulary that doesn't contain the letters "ş" and "ţ" (Dumitrescu et al., 2020)

Another model is XLM Roberta by Alexandra Ciobotaru that is based on xlm-roberta-base (Ciobotaru). Regarding XLM-RoBERTa, this model was trained on a very large corpus containing 2.5TB of data in a self-supervised way that enabled the training to be ran on the large amount of data (Conneau et al., 2020).

In order to detect sarcasm, the project started by the analysis of the model put forward by Meng et al. which contains 4 layers: text representation, semantic feature extraction, sarcasm semantic relation and sarcastic intent discrimination (Meng et al., 2023). Below the layers are presented and the relationships between them Figure 2:

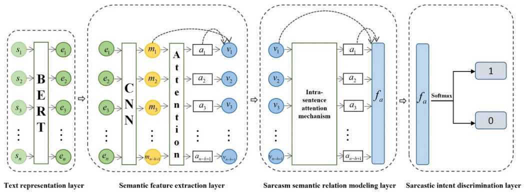
s_1
s_2
s_3
s_n
BERT
e_1
e_2
e_3
e_n
CNN
m_1
m_2
m_3
m_{n-h+1}
Attention
a_1
a_2
a_3
a_{n-h+1}
v_1
v_2
v_3
v_{n-h+1}
Intra-sentence attention mechanism
f_a
Softmax
1
0
Text representation layer
Semantic feature extraction layer
Sarcasm semantic relation modeling layer
Sarcastic intent discrimination layer

Figure 2: Layers of the reference model (Meng et al., 2023)

We can clearly observe that the model starts by processing the data using BERT which provides 768 inputs to a convolutional neural network with one layer with the purpose to break the phrase structure into multiple semantically rich phrase fragments (Meng et al., 2023). The attention layer is inspired by the one put forward by Amir et al. and has the purpose of capturing both the content and the context relevant aspects that can be found in the analysed text (Amir et al., 2016).

After this step, the output of the second layer is sent to the third which models the semantical sarcasm relations with the purpose of noticing semantical contradictions or emotional contradictions between the phrase characteristics in order to detect sarcasm (Meng et al., 2023).

The last layer uses the output of the third in order to simply apply the softmax function and discriminate if the text is sarcasm or not (Meng et al., 2023).

3 Method

This paper has already discussed the Meng et al. model. While using an attention mechanism, this model may add unnecessary complexity to the problem.

Therefore we will base our models on it by using the following changes:

1. First, we will completely remove the attention mechanism and instead add a second convolutional layer to the CNN. The paper hopes that this will enable to capture the non-linearity of the data while excluding the attention mechanism. Obviously a BERT model will be used.
2. Secondly, we will adapt the attention mechanism to the first model and compare the results. If the assumption is right, then the second model will behave poorly. Obviously a BERT model and a 2 convolutional layers CNN will be used but parts of the CNN may be adjusted to fit the attention mechanism.

When it comes to choosing a BERT model, the paper previously analysed two Romanian language models. Both of the models will be using Alexandra Ciobotaru's BERT model as it is more complex and can better represent the data.

In order to use the chosen BERT model with the CNN we will need to do some permutations and be sure that the correct data will be inputed in the CNN.

Digressing from SaRoCo fine-tuned Ro-BERT (Rogoz et al., 2021), we will also use RoBERTa which will then be used in a two convolutional layer CNN.

Therefore this paper aims to compare itself with the SaRoCo fine-tuned RO-BERT and Meng et al. CNN with attention mechanism. Therefore both models will be compared with each baseline.

3.1 Dataset

The used dataset belongs to the Nitro NLP 2024 competition (Cristi Bleotiu) and is similar to a part of SaRoCo which is considered the state of the art dataset for Romanian language satire. It is composed of satirical and non-satirical articles from Romanian news websites (Rogoz et al., 2021).

By analysing the dataset, we can see that the titles have a limited numer of characters Figure 3 while the content contains outliers with large number of characters Figure 4.

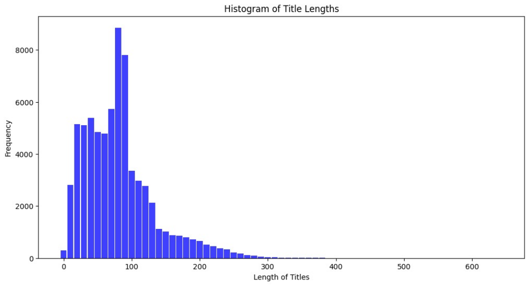

Figure 3: Visualization of title length

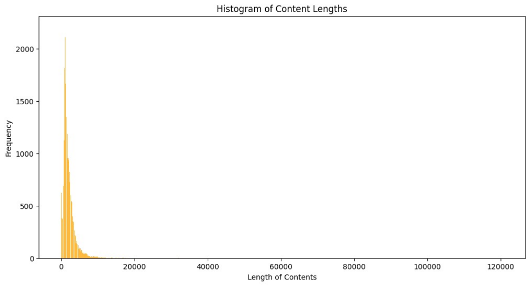

Figure 4: Visualization of content length

In order to visualize the data without the outliers, we have limited the character length frequency and excluded those below 50 appearances Figure 5.

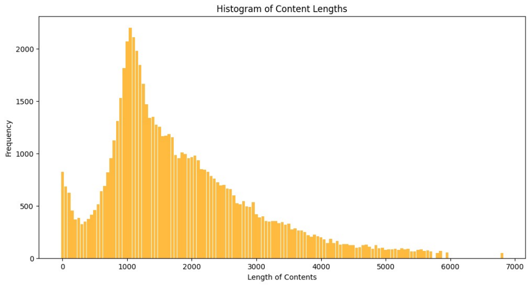

Figure 5: Visualization of content length without outliers

When it comes to the number of articles in SaRoCo, this has 55,608 articles out of which 27,628 are satirical and 27,980 are not (Rogoz et al., 2021). Therefore we see that the dataset is balanced. One of the basic assumptions would be that sarcasm is an outlier in real life communication but this has to be checked in a more comprehensive dataset.

3.2 Preprocessing

The only preprocessing done is to replace the letters "ș" and "ț" in every batch of test both in the training data and in the test according to (Dumitrescu et al., 2020).

We have trained the models with and without this preprocessing technique.

3.3 Models

Both models will have the text tokenized and truncated to a 512 limit. If the limit is not reached, padding will occur.

The first model is a CNN with 2 convolutional layers that processes RoBERTa's output. This model has the following steps:

1. Take the input_ids and attention_mask from the previous step and input them into RoBERTa. The fact that the tokenization was done with RoBERTa's tokenizer will ensure that it will be compatible with this step.
2. Take RoBERTa's output and only keep the last hidden state. This will ensure that the data is relevant to the CNN.
3. In order to make the data compatible with the CNN, it will need some permutations. Therefore, the three dimensions will be switched by the following rule: the first dimension remains unchanged, the second dimension will be switched with the third.
4. Because RoBERTa will output an array of 768 elements, the first convolutional layer will downsize that to 256.
5. Rectified Linear Unit will be used as an activation function after the first convolutional layer
6. In the second convolutional layer the number of nodes will be further downsized from 256 to 128.
7. Rectified Linear Unit will be used again as an activation function after the second convolutional layer
8. An adaptive max pooling will be used in order to reduce the spatial dimension to 1
9. The last layer is a dense layer that will ensure the previous layer is fully connected in order to preserve non-linear relations
10. In order to discriminate sarcasm the model uses the maximum of the sigmoid function on the last dimension. This diverges from Meng et al. but it is used to further preserve the non-linear relations

The first model shows an 87.612% balanced accuracy, way above the 49.7% (Cristi Bleotiu) required balanced accuracy.

The second model builds upon the first and enhances it with Meng et al. attention mechanism.

When it comes to the use of RoBERTa, this second model uses the same processing as the first, therefore an in-depth analysis is not required since it has been already written in detail above. What concerns the second model will be written below.

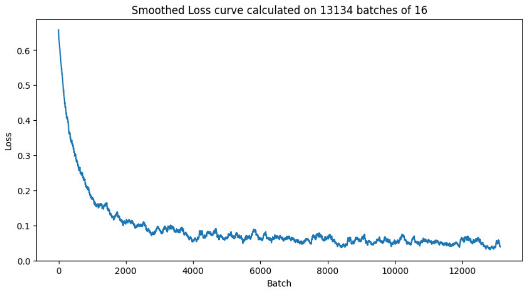

Figure 6: Loss graph calculated on every batch

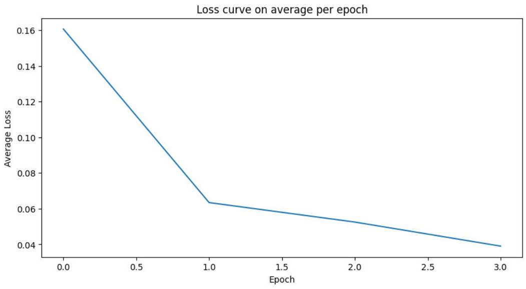

Figure 7: Loss graph calculated on every epoch

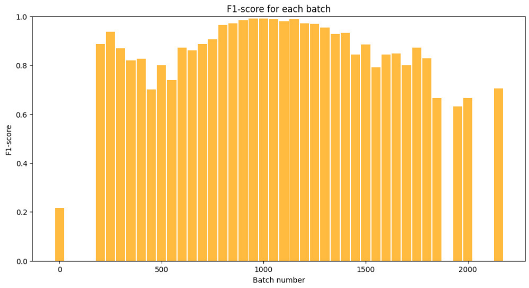

Figure 8: F1 score calculated on SaRoCo

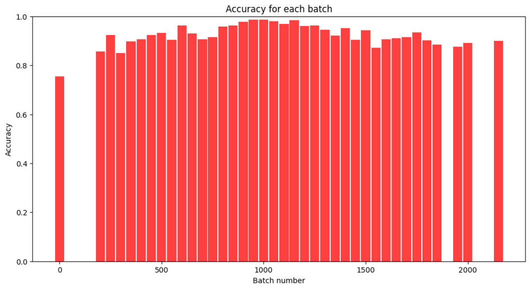

Figure 9: mAP score calculated on SaRoCo

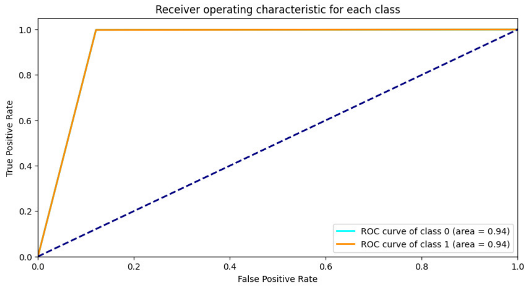

Figure 10: ROC-AUC score calculated on SaRoCo

1. The first CNN layer remains the same, the reason being the 768 item array returned by RoBERTa.

2. Rectified Linear Unit will be used again for activation.

3. In order to simplify the CNN, a 0.5 dropout will be used. This wasn't present in the first model.

4. A second convolutional layer will be added but this time it will upscale the number of nodes from 256 to 768. Therefore, this model will use a bottleneck made with convolutional layers. This will also enable a quick adaptation of Meng et al. attention layers.

5. After this layer Rectified Linear Unit will be used for activation.

6. Another dropout layer is used in order to get rid of redundancy that can impact the performance of the model.

7. An adaptive max pooling will be used to reduce the spatial dimension to 1 just like in the previous model.

8. Phrase attention layer comes next which ties the embeddings for each text with a dense 768 node layer and applies the hyperbolic tangent for activation, after which it will multiply the result with a tensor of shape (768, 1) initialized with the Xavier uniform distribution, after which a log_softmax on the first dimension will be used and the result returned in accordance with Meng et al. phrase attention layer (Meng et al., 2023).

9. The self attention layer comes right after and takes as a parameter the result of the phrase attention and text embeddings matrix multiplication, after which it constructs a matrix in order to compute the magnitude of the semantic conflict between each word, followed by a maxpooling in order to determine the largest attention score for the current phrase (Meng et al., 2023).

10. In order to use the result from Meng et al. in the current model, a permutation of the self attention result for the current batch has to be made. It will first concatenate the tensor on the third dimension, then permute in the

following order: the third dimension becomes first, the frist becomes second, the second becomes third and the fourth dimension remains unchanged.

11. After this, max pooling to the first spatial dimension will be used and a dense layer like in the previous model will be added.

12. The tensor will have the second dimension excluded.

13. The sigmoid function will be used just like in the previous model.

The second model shows a 63.535% balanced accuracy, a lot below the first model, but still above the baseline.

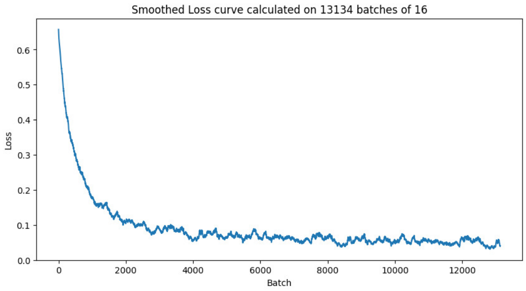

Figure 11: Loss graph calculated on every batch

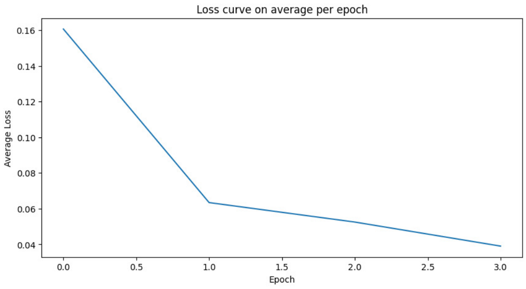

Figure 12: Loss graph calculated on every epoch

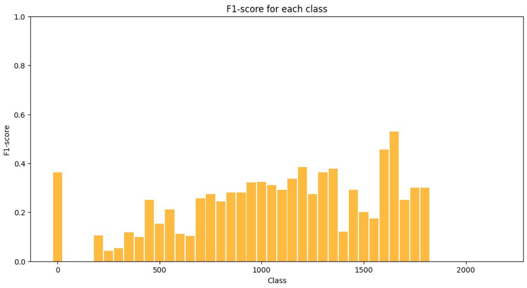

Figure 13: F1 score calculated on SaRoCo

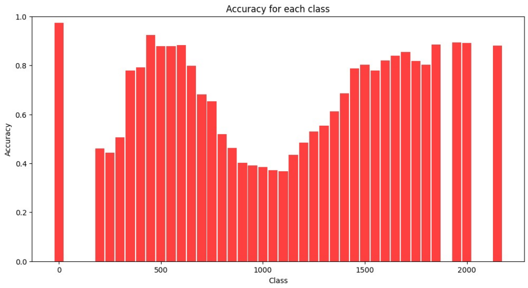

Figure 14: mAP score calculated on SaRoCo

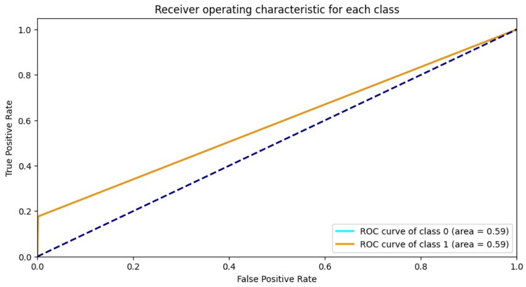

Figure 15: ROC-AUC score calculated on SaRoCo

We can now clearly observe that the attention model, despite being more complex, performs worse compared to the model that uses a fully connected convolutional network to analyze the embedings produced by a BERT model. Let's analyze the graphs. All the scores were calculated on the SaRoCo dataset, for consistency. The ROC-AUC curve Figure 10 and Figure 11 is used to show the relation between true positive and false positive rates. The greater the area under the graph, the better the accuracy. For the model used in the actual competition, the area is considerably big: 97%. The model using an attention mechanism scores considerably lower, at 67%.

Now turning to the F1 and mAP scores Figure 13 and Figure 14, we can see that the trends are the same. The model with the attention mechanism has a low F1 score overall. The mAP score is a bit higher, but it fluctuates a lot across the batches, having a precision of almost 1 on some batches, while others are below 0.5, showing that the model is not consistent. On the other hand, the plain model has an incredibly good F1 and mAP score over all batches, with small exceptions, that can be neglected.

The Loss curve Figure 11 and Figure 12 is pretty steep for the plain model, which can be a sign of overfitting (the loss is 0.06 after the first epoch, which is worringly low for the first epoch).

4 Conclusion

It is clear that the balanced accuracy of both models are modest but an observation can be made that the initial hypothesis which was to render the model better by using a second convolutional layer in the CNN and excluding the attention mechanism holds true. This means that the attention mechanism performs worse than a convolutional layer that downsizes the number of nodes and can be replaced by it.

5 Future Work

As a main general directions for future work is the addition of text preprocessing that can better improve the model.

Also, Meng et al. attention mechanism is not rendered useless by the first model but another comparison on preprocessed data is needed because it can be influenced by distracting words.

Another direction is a comparison between this model and a similar model using another BERT model.

Ethical Statement

As it happens in sarcasm detection, one can use it in order to enact censorship of dissenting opinions. This can be employed either as a private censorship solution or even backed by an online platform or a nation state. As with all NLP models, ethical use is encouraged.

Limitations

The main limitation is that the comparison with SaRoCo could not have been done due to the small dataset used. Also, the comparison with the original model could not have be done since it uses a different BERT model. The only comparison in this article is with the Nitro NLP baseline. Therefore testing the models on the full SaRoCo dataset is needed.

Acknowledgements

We want to thank the University of Bucharest for hosting Nitro NLP 2024 and Codiță for orange cat debugging.

References

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

Ashwitha Anup, Shruthi Gowda, Shruthi R, Makarand Upadhyaya, Abhra Ray, and Tc Manjunath. 2020. Sarcasm detection in natural language processing. *Materials Today: Proceedings*, 37.

Alexandra Ciobotaru. Alegzandra/xlm-roberta-base-finetuned-on-redv2 · hugging face. Visited on 2024-05-13.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Miruna-Andreea Zavelca Cristi Bleotiu, Lucian Istrati. Nitro nlp. (2024). nitro language processing - 3rd edition - satire.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Stefan Dumitrescu, Andrei-Marius Avram, and Sampo Pyysalo. 2020. The birth of romanian bert. pages 4324–4328.

Srijita Majumdar, Debabrata Datta, Arpan Deyasi, Soumen Mukherjee, Arup Bhattacharjee, and Anal Acharya. 2022. Sarcasm analysis and mood retention using nlp techniques. *International Journal of Information Retrieval Research*, 12:23.

Jiana Meng, Yanlin Zhu, Shichang Sun, and Dandan Zhao. 2023. Sarcasm detection based on BERT and attention mechanism. *Multimed. Tools Appl.*, 83(10):29159–29178.

Bleau Moores and Vijay Mago. 2022. A survey on automated sarcasm detection on twitter.

Ana-Cristina Rogoz, Mihaela Gaman, and Radu Tudor Ionescu. 2021. Saroco: Detecting satire in a novel romanian corpus of news articles.

Cate Watson. 2011. Notes on the variety and uses of satire, sarcasm and irony in social research, with some observations on vices and follies in the academy. *Power and Education*, 3:139.