Sarcasm Detection for Inclusive Writing

Raffaele Iaquinto

Simon Iyamu Perisanidis

ETH Zurich

Building a Robot Judge: Data Science for Decision-Making HS2021

Course Project

In collaboration with Witty Works AG

Abstract

In the context of the course Building a Robot Judge and in collaboration with the company Witty' Works, our project's purpose is to develop an NLP model that can detect sarcasm in English text. Witty' Works, which is a startup that is building an assistant tool for inclusive writing, can benefit from integrating our model into their pipeline to improve the accuracy of their sentiment analysis tool. In this work we exploit the transfer capabilities of BERT and fine-tune the task of sarcasm detection. We also experimented with a recent and very promising derivation of BERT, called MPNet. Our results get close to the baseline for accuracy and recall but show large deviations for the precision and F1 score. Those deviations can be explained by the difference in model validation datasets. Further, we found that the main driver of the low precision score of our models is probably the high amount of false negative labels in the validation dataset. Our solution was shared with Witty.works and can be utilized into their inclusive writing assistant pipeline.

Keywords: NLP, sarcasm detection, sentiment analysis

Contents

Introduction

Motivation

The goal of our project is to create an NLP model that is able to detect sarcasm in English text. Such a model could be used as a tool for improving written communication, both from a content consumer as well as a content creator perspective. It could be used by social media platforms in order to make readers more aware of sarcastic contributions and prevent misunderstandings. Or it could be integrated into writing tools to help authors write more clearly. For this project we will be collaborating with the company Witty.works which is developing such a tool.

In order to make a case for the fact that sentiment analysis models perform poorly in sarcasm detection, we did a small experiment. We have randomly selected 5 clearly sarcastic reddit comments, and inferred their sentiment using a state-of-the-art BERT-based sentiment classifier. As it can be seen in Table 1, all 5 example sentences were falsely classified as positive.

Table 1: Example of misclassification of sarcasm

Sarcastic input sentence	Label	Score
oh wow i am so surprised i never saw this coming.	Positive	0.9978
good luck with that.	Positive	0.9998
i am glad we have priorities straight.	Positive	0.9947
there's no way this could go wrong.	Positive	0.9993
good thing we have religion bringing peace into the world.	Positive	0.9997

Background

Sarcasm detection is the task of detecting verbal irony when given a text as input. Since deciding whether a sentence contains sarcasm or not is subjective and not clearly defined, sarcasm detection is exceptionally difficult for machines.

Transfer Learning is a method in machine learning, where deep models that are trained using big sized training datasets, are readjusted to solve a different but similar task. The benefits of transfer learning is that it can achieve satisfying results even when the new training is relatively small. It is widely used in a variety of tasks in deep learning. In this work, we will use transfer learning, from variations of BERT to perform sarcasm detection.

Literature review

Approaches to automatically detect sarcasm can be broadly split into two categories, content and context based methods. Content based methods use features about the local content of a text passage to identify sarcasm. Most papers in this category use Support Vector Machines (SVMs) and k-Nearest Neighbor (kNN) algorithms for classification. Carvalho et al. (2009) and Davidov et al. (2010) use acronyms, adverbs and word frequencies. Ravi and Ravi (2017) use statistical and semantic features and both Ling et al. (2016) and Farias et al. (2018) use sentiment and sentence structure related features. More recent papers started using artificial neural networks to detect sarcasm, for example Zhang et al. (2019) and Huang et al. (2017) both use Recurrent Neural Networks (RNNs) with Word2Vec embeddings and contextual information.

When trying to identify sarcasm, context is very important. Depending on where a sentence is placed in the whole text its meaning can change from non-sarcastic to sarcastic. Therefore, using methods that incorporate contextual information can significantly improve sarcasm detection

performance. One tool that incorporates context are language models, which will be used in this paper as well and will be discussed in depth in the methods section, that are trained on a large text corpus with masked and permuted language modeling to extract connection between words. Potamias et al. (2020) use such models to detect sarcasm.

Besides modeling techniques, data is the second component needed to create a sarcasm classifier. The three data collection methods used to create a dataset for sarcasm classification are distant supervision, manual labeling and self labeling. Distant supervision uses rules to label sarcastic texts (Ptacek et al. (2014), Khodak et al. (2018) and Barbieri et al. (2014)). Examples of this are Twitter comments tagged with #sarcasm or words related to sarcasm. Abercrombie and Hovy (2016) and Joshi et al. (2016) tasked human annotators to manually label previously collected text samples, whereas Opera and Magdy (2019) used the self labeling method to create a dataset by asking people to send them self-annotated text samples. Each method comes with advantages and disadvantages. Distant supervision scales well, as retrieving a large number of text samples from social media platforms is relatively cheap, but it leads to more noise in training data. Manually labeling text is quite expensive and slow but allows for better quality control.

Data

We used the Self-Annotated Reddit Corpus (SARC) to train our Natural Language Processing (NLP) model to detect sarcasm in text. The corpus has been created and made available by the Princeton University Natural Language Processing group. Khodak et al. (2017) collected 533 million comments, of which 1.34 million are annotated as sarcastic, that have been written between 2009 and 2017 on the online discussion platform reddit.com. All sarcastic comments are self-annotated. When submitting a comment on reddit.com the author has the option to add the annotation "/s" at the end of the comment in order to mark the comment as sarcastic. For each comment the dataset contains its post id, parent comment (if it is a reply to another comment), the author of the comment, the subreddit (the general topic of the comment), the score of the comment (determined by votes of platform users), the date the comment was made and a sarcasm label.

We use the provided filtered version of the dataset that is composed of 8.44 million labeled comments, of which 224.3 thousand are sarcastic. The raw dataset is filtered by only selecting comments from authors that have previously used the sarcasm annotation option and that have written the "/s" at the end of their comments. This technique is used to reduce the number of false positives, comments that are incorrectly labeled as sarcastic, in the dataset. To reduce the number of false negatives, sarcastic comments that are not labeled as such, all comments that follow a sarcastic comment are removed.

The filtered dataset contains over 7900 different subreddits, many of which are about niche topics that are of little interest to a general audience. Given that sarcasm relies heavily on context and our goal is to create an NLP model that recognizes sarcastic comments of a general

audience we select a subset of the original dataset for model training. By selecting only subreddits containing more than 5000 comments we are left with 123 subreddits that contain 90% of all comments. We further exclude subreddits that in our view are still too niche, such as subreddits on video games and sports teams, and aggregate the remaining subreddits into 12 categories we think are most relevant to detect sarcasm in text written by a general audience. In a last step we go through the subreddits that have more than 1000 comments (making up over 95% of all comments) and add the comments of the subreddit to the general categories if appropriate. We end up with 38 subreddits, aggregated into 10 general categories that contain a total of 1.6 million (44.4k sarcastic) comments.

Table 2: Subreddits by category

Category	Subreddit
advice	LifeProTips
economics	Economics, business, MapPorn
entertainment	Music, Documentaries, books, hiphopheads, television
news	worldnews, reddit.com, news
politics	politics, ukpolitics, Libertarian, worldpolitics, Conservative, PoliticalHumor, hillaryclinton, SandersForPresident
regional	australia, canada, unitedkingdom, india, europe, european
science	science, environment, space
social group	MensRights, TwoXChromosomes, BlackPeopleTwitter
sports	sports, formula1, soccer
technology	technology, gadgets, Futurology

When comparing the comments across categories the average length of comments in the sports category is below average length, comments in the science category are longer than

average and the other 8 categories are close to the average. It does make sense that discussing more complex topics like science requires longer comments compared to sports discussions. When comparing sarcastic to non-sarcastic comments we see similar lengths but lower standard deviations of lengths across categories.

Table 3: Summary Statistics

Category	Number of total comments	Percent of sarcastic comments	Mean comment length		Std. dev. comment length		Min comment length		Max comment length	
			Non-sarc.	Sarc.	Non-sarc.	Sarc.	Non-sarc.	Sarc.	Non-sarc.	Sarc.
News	754743	1.81%	63.87	61.26	51.62	39.52	1	2	9985	751
Politics	422882	3.58%	64.54	65.90	51.24	38.84	1	3	4676	518
Tech	119018	1.60%	67.63	59.58	57.56	41.00	1	2	9822	692
Sports	90164	3.65%	48.95	48.81	40.22	38.22	1	3	1576	1318
Entertain.	72731	1.69%	55.68	64.35	60.14	41.30	1	4	9614	334
Regional	61737	10.62%	62.82	64.35	50.94	42.81	1	3	1861	349
Social group	20835	6.75%	62.33	51.99	72.36	34.07	1	3	7476	342
Science	16305	3.05%	78.99	58.27	50.79	35.59	1	2	799	299
Economics	8881	7.01%	58.08	67.23	43.78	40.60	1	6	783	329
Advice	6076	2.19%	65.01	57.26	44.25	34.13	4	7	674	174
Total	1573372	2.82%	63.17	62.13	51.88	39.57	1	2	9985	1318

Methods

In this section, the main methods of our approach are documented. In particular, the preprocessing techniques used to prepare our data for our machine learning model, the architecture of the model itself and the details of the training procedure. For the implementation of our experiments we used Hugging Face, an open source NLP library where developers can build, train and deploy state of the art NLP models.

Preprocessing

Sarcasm Detection is considered to be a classification problem. Although our dataset contains text, machine learning models can only have integer numbers as input. Therefore, in order for our model to exploit the information of each record of the training data, the first step is to extract and select text features. There exist multiple techniques to tackle this task many of which we have experimented with during our research. The methods that extract the most useful features for our specific classification problem are summarized in the following paragraphs.

Cleaning the data. Removing unnecessary symbols, hashtags or web links can be a beneficial step. In our case, punctuation can play an important role in determining whether the input is sarcastic or not, so we did not remove many symbols. Therefore, in this step we only remove mentions (words starting with @) and links (words starting with http). Also, we convert all letters to lowercase, so that words that contain capital letters are considered the same as words that do not.

Tokenization. After our data is clean, tokenization can be used to split each sentence into a list of smaller units such as individual words or terms. Each of these units are called tokens. This is a crucial step in our effort to transform each sentence to a list of integers.

All these adjustments to our data can be done easily with the help of the Hugging Face tokenizers. One can simply select a tokenizer and their preferred hyperparameters, without worrying about how to implement it exhaustively.

Architecture

In our work, we have mainly experimented with two models. The first one is BERT(Bidirectional Encoder Representations from Transformers) , one of the most popular pretrained NLP models that is used for transfer learning, which has achieved state-of-the-art results in multiple tasks. Second, we tried MPNet, a pretrain NLP model that derived from BERT in order to improve some of its limitations. This method has outperformed the original BERT in a plethora of tasks, and therefore it is interesting to test whether it is also better than BERT on sarcasm detection.

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based machine learning model that was developed by researchers at Google AI Language in 2018. It has presented state-of-the-art results in a wide variety of NLP tasks, like Question Answering, Natural Language Inference, and others.

BERT's most fundamental breakthrough is the use of the bidirectional training of Transformer, a popular attention model, to language modeling. In contrast to earlier research, which looked at text sequences from left to right or a combination of left-to-right and right-to-left training, this study proves that a language model which is bidirectionally trained can have a deeper understanding of language context and flow. The researchers have developed a novel method called MaskedLM which allows bidirectional training in models in which it was previously impossible.

With the aim to use a pre-trained BERT model and fine-tune it to our task of sarcasm detection, we load a smaller version of the pre-trained BERT, which is called **DistilBert**, from the open source library Hugging Face. BERT was pre trained on BookCorpus, a dataset consisting of the English Wikipedia and 11,038 unpublished books. In particular, it was trained on 4 cloud TPUs in Pod configuration (16 TPU chips total) for one million steps (with a batch size of 256) and with the use of the Adam optimizer. The sequence length was limited to 128 tokens for 90% of the steps and 512 for the remaining 10%. To reduce the size of BERT and make fine–tuning faster, DistilBert is trained with a process called knowledge distillation during the pre-training phase. This reduces the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster.

MPNet. BERT has proven to be exceptional in a plethora of NLP tasks, but suffers from a few limitations. Namely, BERT neglects the dependency that exists among predicted tokens and position information of a sentence. MPNet was proposed by researchers from Microsoft Research Asia in 2020 in order to make up for those weaknesses. It leverages the dependency between predicted tokens through a method called permuted language modeling. Additionally, in order to reduce the position discrepancy, it takes auxiliary position information as input to make the model see a full sentence. Thus, MPNet has outperformed previous state-of-the-art pretrained methods (e.g. BERT, XLNet, RoBERTa) on a variety of tasks.

Similarly, with the aim to use a pre-trained MPNet model and fine-tune it to our task of sarcasm detection, we load it from the open source library Hugging Face. The model was pre trained on a similar corpus with BERT, of size 160GB. The length of sentences in each batch is

limited to 512 and a batch size of 8198 sentences is used. The total training procedure took 35 days, using 32 NVIDIA Tesla 32GB V100 GPUs.

Training

For each model we follow the same process. First we download the pretrained model and weights using HuggingFace. Our goal is to perform transfer learning on the model. Therefore, the head of each model is initialized randomly and we fine-tune it so that it adapts to our specific problem. We use Adam optimizer with a learning rate of 4e-6, and we train our models for 3 epochs and a batch size of 8 for MPNet and 16 for DistilBert. We conducted our experiments on Google Colab environment, with an NVIDIA Tesla K80 GPU.

One issue we face is that the dataset for training contains only around 2.8% sarcastic comments, it is therefore highly unbalanced. To balance our dataset for training we use downsampling. We first split the data into training and testing sets using a 99%/1% split (after downsampling the split will be 85%/15%) and then only apply downsampling to the data used for training. The sampling works as follows. For every sarcastic comment per post id, defined in the previous section, we select a non-sarcastic comment (from the same category) at random. Since the training set contains around 35k sarcastic comments, this method yields a final dataset of around 70k comments, evenly split between sarcastic and non-sarcastic comments, that can be used to train the NLP model.

Evaluation

Metrics

Before presenting the results themselves, it is necessary to briefly explain the metrics used. In order to evaluate the effectiveness of the classification task throughout our experiments we used Accuracy, Precision, Recall and F1-Score.

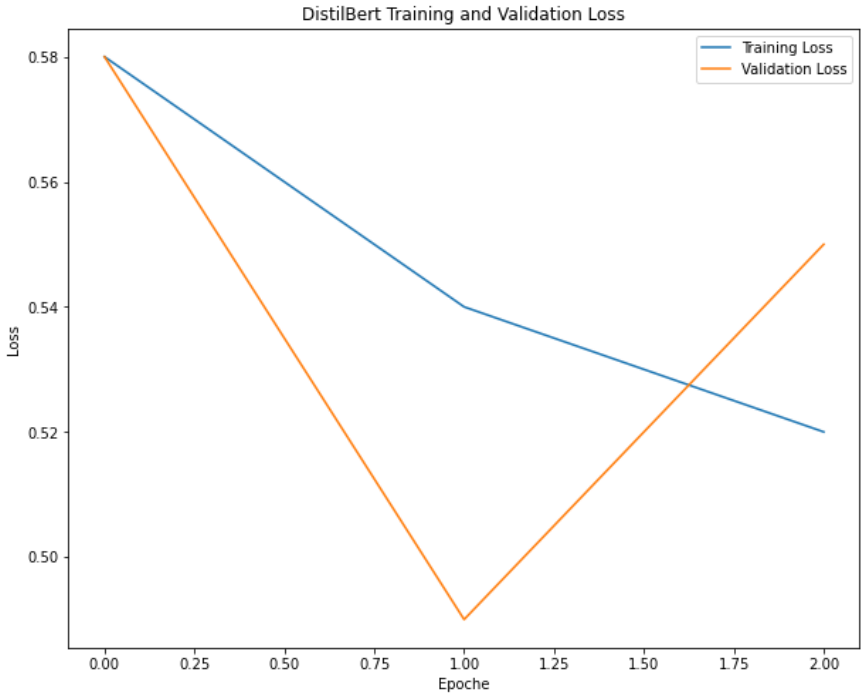
The F1-Score is the harmonic mean of Precision and Recall. When we want to increase the precision of our model, recall decreases and vice-versa, therefore the f1-score is a helpful metric since it captures the trends between them in a single value.

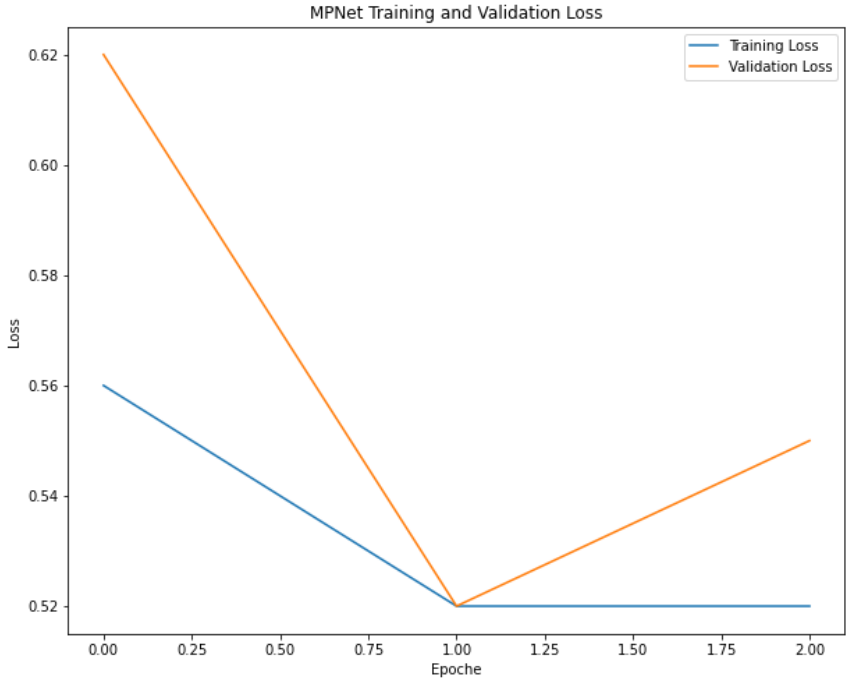
$$f1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

Results

The results we achieved can be seen in Table 4. We chose the results achieved by Potamias et al. (2020) as a baseline for our models. They trained a RCNN–RoBERTa model that produces state of the art results as a sarcasm classifier and tested it on a subset of SARC that only contains comments related to politics. The scores for DistilBERT and MPNet models were achieved in the second epoch of training. In the next epoch our model started overfitting (see Figure 1).

Figure 1. Loss by Epoch



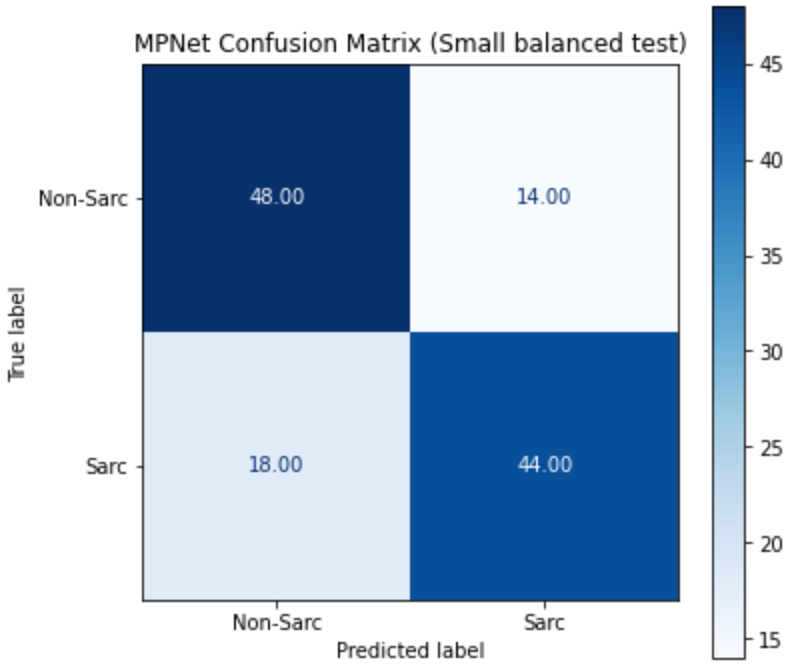


Both models get close to the accuracy and recall values of the baseline but miss on precision by an order of magnitude and have an F1–Score that is around 6 times lower. A potential explanation for the large deviations in precision and F1-Score can be found in the different model validation methods. Potamias et al. (2020) used a balanced subset of SARC for both training and testing their model, whereas we left the dataset for validation imbalanced. To confirm our hypothesis we sampled 62 non-sarcastic and 62 sarcastic comments at random from the dataset that was set aside for testing and used the data to validate our MPNet model. The results shown in Figure 2 confirm our hypothesis that the difference in precision between our models and the baseline come from different validation setups. In our opinion the models should be validated on data that is as close to the real world as possible. Since sarcasm is quite rare compared to non–sarcastic text any robust sarcasm classifier that is intended to be deployed in production systems should be able to perform well on highly imbalanced data. Therefore, we will further investigate the reasons for the low precision score and how it could be improved.

Table 4. Classification Metrics

	Baseline (Potamias 2020)	DistilBERT	MPNet
Accuracy	0.79	0.79	0.77
Precision	0.78	0.08	0.08
Recall	0.78	0.65	0.71
F1-Score	0.85	0.14	0.15

Figure 2. MPNet Confusion Matrix of balanced test set



Figures 3 and 4 show the confusion matrix for both models validated on the imbalanced test dataset that contains a total of 15734 comments, of which 444 (2.8%) are labeled as sarcastic. From the confusion matrices as well as the ROC curves shown in Figure 5 we can conclude that both models are able to distinguish between classes (sarcastic vs non-sarcastic), with a recall score of over 0.7, but the likelihood of a sarcastic comment actually being sarcastic when classified as such is very low. In other words, the low precision score of both models is mainly caused by classifying non-sarcastic comments as sarcastic. To investigate our models' predictions a little bit further we sorted the validation samples by the model loss. The predictions with the highest loss values are mainly composed of miss predictions of non-sarcastic comments, i.e.

models wrongly predicted sarcasm. The 40 predictions with the highest loss values for MPNet are only of this kind. When looking at the content of the comments more closely a pattern emerges. Out of the top 40 comments for both models (which have significant overlap) we would classify 75% percent as sarcastic. Even if this is a subjective measure (see Table 5) we believe that this pattern is a strong indication for a significant amount of false negative labels (sarcastic comments that are labeled as non-sarcastic) in the dataset. The self–annotation of comments by the commenter seems to introduce a high amount of noise. This observation suggests that the true precision (without the negative influence of miss labeled data points) of our models might be significantly higher. In conclusion, our results get close to the baseline for accuracy and recall but show large deviations for the precision and F1 score. As we have shown above, those deviations come from different model validation datasets. Further, we found that the main driver of the low precision score is probably the high amount of false negative labels in the validation dataset.

Figure 3. DistilBert Confusion Matrix of imbalanced test set

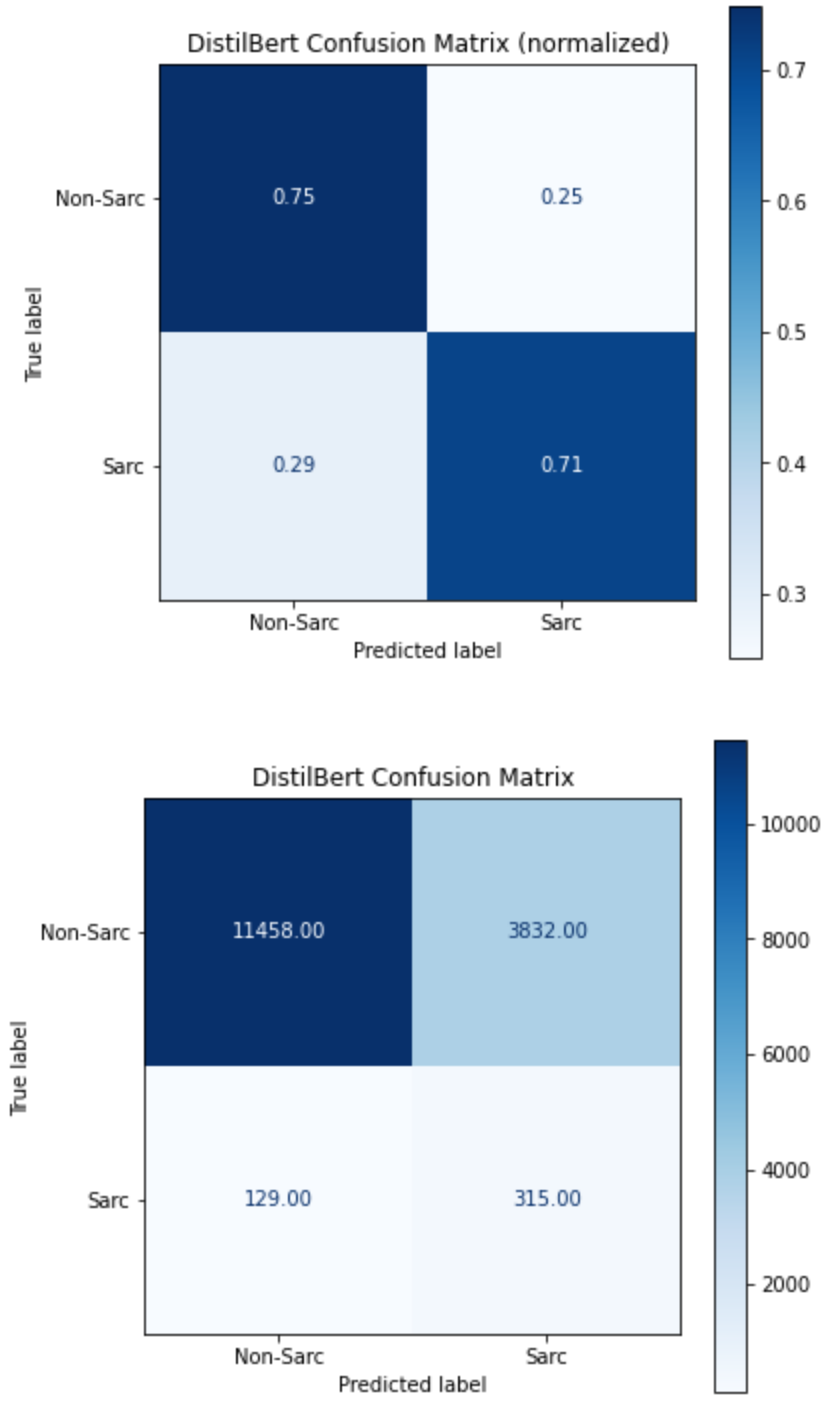


Figure 4. MPNet Confusion Matrix of imbalanced test set

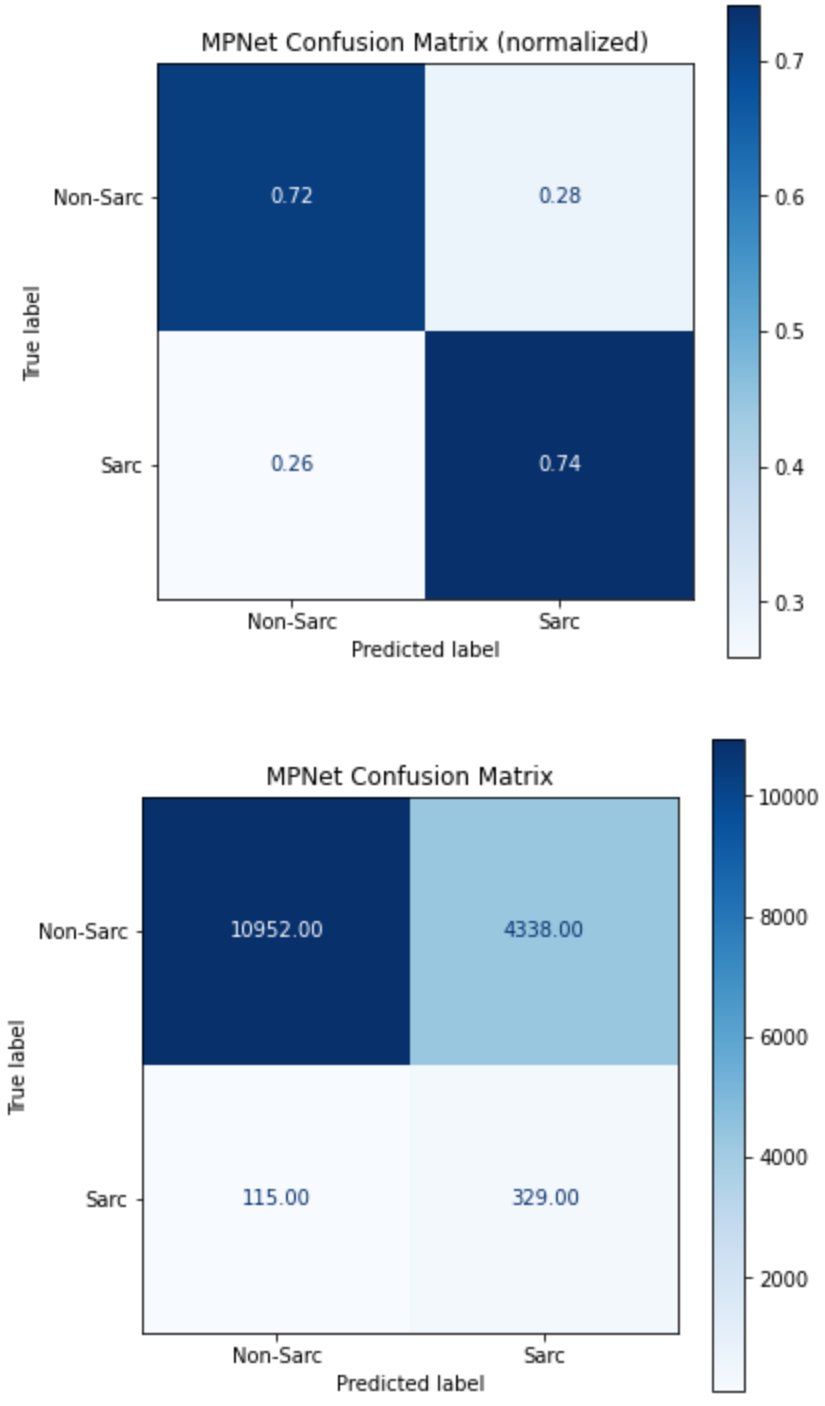


Figure 5. DistilBert and MPNet ROC curve

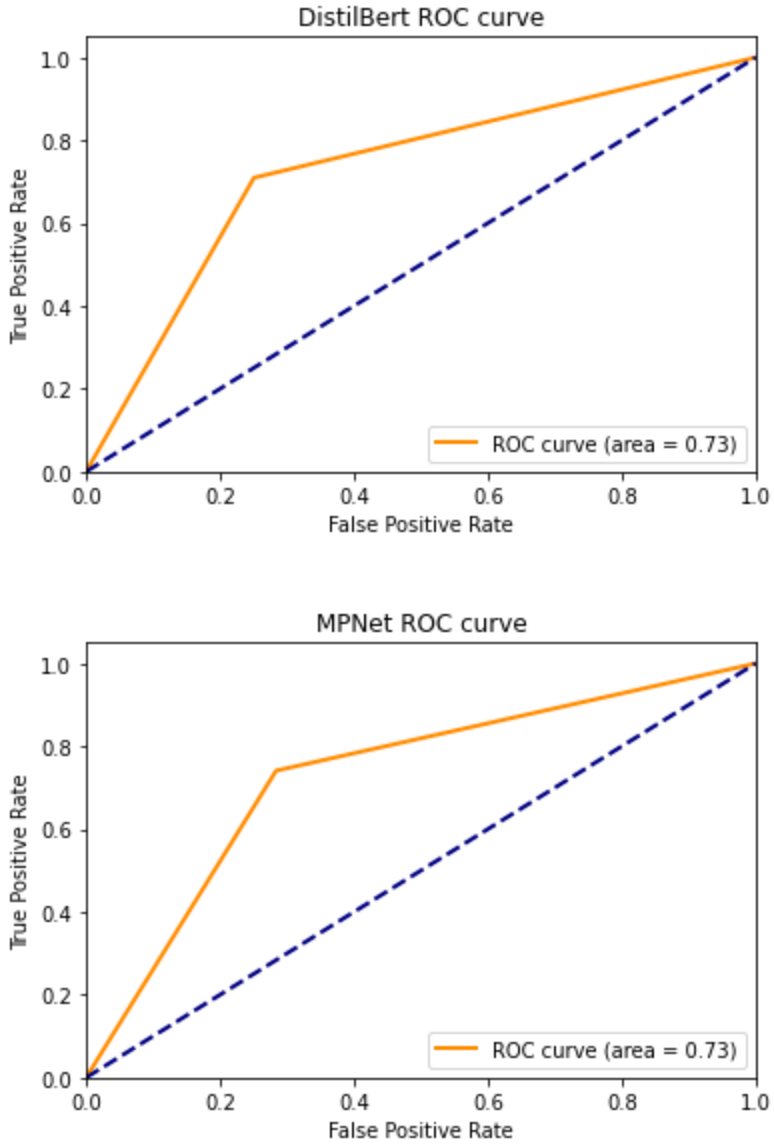


Table 5. Comments labeled as non-sarcastic but classified as sarcastic

Post	Comment
States Consider Increasing Taxes for the Poor and Cutting Them for the Affluent	Yeah that'll fix wealth inequality
Bill Regulating 3D Printed Guns Announced In NYC -- The bill would make it illegal to use a 3D printer to create any part of a firearm unless the person is a licensed gunsmith	Sure, cause we all know that the law will stop the criminals cold!
Trump signs internet privacy repeal	I'm sure this is what all those rural working-class voters wanted him to prioritise when he got into office.
Obama administration moves forward with unique internet ID for all Americans	There's absolutely no way that this could go terribly wrong.
N. Korea has just vowed actual military actions against the US	Oh, now we're worried.
Massachusetts police used a military style helicopter to seize a single marijuana plant from an 81 year old woman using it to ease her arthritis and glaucoma.	As long as they didn't go overboard
Student Drug Informant Found With a Bullet in His Head and Rocks in His Backpack	Its all in a day's work for our public servants.
NBN Co CEO Bill Morrow's pay rises 18.7 per cent to $3.6 million	This money *could* have gone into funding prostate cancer awareness.
New data shows 85% of humans live under a corrupt government	After this shocking revelation, the wheater
Hong Kong has too many poor people to allow direct elections, leader says	Well duh, I mean, people without money *obviously* can't have opinions!
Comcast to buy Time Warner Cable in all stock deal	Yeah this will be great for competition within the industry
Costco will again stay closed on Thanksgiving this year, bucking the trend of retailers opening their doors earlier and earlier: "We simply believe deserve the	But I am sure they are just hating all this free publicity they are getting from this.

opportunity to spend Thanksgiving with their families	
Schwarzenegger responds to Trump: 'Why don't we switch jobs?'	Yeah, because Arnold did such a bang up job being the Governor of California.
Kim Jong Un Orders Military To Kill South Korean Leaders	But Dennis Rodman says he's a good guy!

Conclusion

Our results show that language models are able to distinguish sarcastic comments from non-sarcastic ones. However, there is still room for improvement. Two main avenues for future research on sarcasm detection are better data labeling techniques and hyperparameter tuning.

Our models detected some potentially mislabeled comments in the validation dataset we used, which leads us to conclude that our training dataset might also contain a large number of false negatives that lead to worse model performance. Coupled with the fact that in the real world the occurrence of sarcastic comments is already very rare compared to non-sarcastic comments, we believe that the search for techniques that allow us to collect correctly labeled sarcastic comments at scale for model training could have a large positive impact on the whole field.

Further improvements to the performance of our models could be achieved by changing some model parameters. Due to hardware limitations we trained our models with a batch size of 16, an increase to 32 or even 64 might yield better results. Fine Tuning the generic language models on the specialized language of our dataset with masked language modeling before training a classifier might be another avenue worth exploring.

References

Abercrombie, G., & Hovy, D. (2016, August). 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 (pp. 107-113).

Barbieri, F., Saggion, H., & Ronzano, F. (2014, June). Modelling sarcasm in twitter, a novel approach. In Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (pp. 50-58).

Carvalho, P., Sarmento, L., Silva, M. J., & De Oliveira, E. (2009, November). Clues for detecting irony in user-generated contents: oh...!! it's" so easy";-. In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion (pp. 53-56).

Davidov, D., Tsur, O., & Rappoport, A. (2010, July). Semi-supervised recognition of sarcasm in Twitter and Amazon. In Proceedings of the fourteenth conference on computational natural language learning (pp. 107-116).

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

Farías, D. I. H., Montes-y-Gómez, M., Escalante, H. J., Rosso, P., & Patti, V. (2018, October). A knowledge-based weighted KNN for detecting Irony in Twitter. In Mexican International Conference on Artificial Intelligence (pp. 194-206). Springer, Cham.

Huang, Y. H., Huang, H. H., & Chen, H. H. (2017, April). Irony detection with attentive recurrent neural networks. In European Conference on Information Retrieval (pp. 534-540). Springer, Cham.

Joshi, A., Bhattacharyya, P., Carman, M., Saraswati, J., & Shukla, R. (2016, August). How do cultural differences impact the quality of sarcasm annotation?: A case study of indian annotators and american text. In Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (pp. 95-99).

Khodak, M., Saunshi, N., & Vodrahalli, K. (2017). A large self-annotated corpus for sarcasm. arXiv preprint arXiv:1704.05579.

Ling, J., & Klinger, R. (2016, May). An empirical, quantitative analysis of the differences between sarcasm and irony. In European semantic web conference (pp. 203-216). Springer, Cham.

Oprea, S., & Magdy, W. (2019). isarcasm: A dataset of intended sarcasm. arXiv preprint arXiv:1911.03123.

Potamias, R. A., Siolas, G., & Stafylopatis, A. G. (2020). A transformer-based approach to irony and sarcasm detection. Neural Computing and Applications, 32(23), 17309-17320.

Ptáček, T., Habernal, I., & Hong, J. (2014, August). Sarcasm detection on czech and english twitter. In Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers (pp. 213-223).

Ravi, K., & Ravi, V. (2017). A novel automatic satire and irony detection using ensembled feature selection and data mining. Knowledge-Based Systems, 120, 15-33.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Song, Kaitao, et al. "Mpnet: Masked and permuted pre-training for language understanding." arXiv preprint arXiv:2004.09297 (2020).

Wolf, Thomas, et al. "Huggingface's transformers: State-of-the-art natural language processing." arXiv preprint arXiv:1910.03771 (2019).

Zhang, S., Zhang, X., Chan, J., & Rosso, P. (2019). Irony detection via sentiment-based transfer learning. Information Processing & Management, 56(5), 1633-1644.