niksss at SemEval-2022 Task 6: Are Traditionally Pre-Trained Contextual Embeddings Enough for Detecting Intended Sarcasm ?

Nikhil Singh
Manipal University Jaipur
nikhil3198@gmail.com

Abstract

This paper presents the 10th and 11th place system for Subtask A - English and Subtask A - Arabic respectively of the SemEval 2022 - Task 6. The purpose of the Subtask A was to classify a given text sequence into sarcastic and non-sarcastic. We also breifly cover our method for Subtask B which performed subpar when compared with most of the submissions on the official leaderboard . All of the developed solutions used a transformers based language model for encoding the text sequences with necessary changes of the pretrained weights and classifier according to the language and subtask at hand .

1 Introduction

According to (Yaghoobian et al., 2021), "Sarcasm detection is the task of identifying irony containing utterances in sentiment-bearing texts". Even though sarcastic humor is present throughout social media, it is hard for even humans to comprehend it certainly as the reader doesn't always perceive it the same way the speaker intended with intricate socio-psychological and cultural references. With the way, current models for text representation are trained like Word2Vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014), which have the same representation for a particular word irrespective of the context they fail to incorporate the meaning of sarcastic texts. Even though the recent state-of-the-art language models provide contextual embeddings for words, it still fails to tell the sarcastic humor from normal as we explain later in the paper.

There have been a lot of attempts at computationally automating sarcasm detection in the literature. Discernibly, the task of sarcasm detection can be classified into content and context-based methods. With features like Structural, morphosyntactic and semantic ambiguity features (Reyes et al., 2012),User mentions (replies), emoticons, N-grams, dictionary- and, sentiment-lexicon-based features(González-Ibáñez et al., 2011)) and features based on word embedding similarity (Joshi et al., 2016) coming inside content based method. With the surge of seq2seq based model such as BERT (Bidirectional Encoder Representations from Transformers (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019), etc. have been heavily employed for sarcasm detection in the literature. (Potamias et al., 2020) proposed an R-CNN-RoBERTa which is a hybrid model leveraging RoBERTa's contextual embedding into a recurrent convolutional neural network. (Dadu and Pant, 2020) created an ensemble of RoBERTa and ALBERT (Lan et al., 2019). Seeing the success of pre-trained language model, we decided to leverage the pre-trained contextual embeddings and transformers toward sarcasm detection.

While some of the previous textual sarcasm detection datasets involved annotation via finding some predefined criteria, such as including specific tags (e.g. sarcasm, irony) (Ptácek et al., 2014) associated with the text. Other datasets involved manual labeling (Filatova, 2012) (Yang et al., 2016). However, the mentioned labeling techniques produced noisy or uncertain labels which would further compromise the effectiveness of the models trained on them. Further, most of the sarcasm detection work has been done in the English language and it is highly unlikely that the models trained on one language would generalize well on other languages.

The purpose of SemEval-2022 Task 6 - Intended Sarcasm Detection in English and Arabic (Abu Farha et al., 2022) is to advance the development of automatic textual sarcasm detection by providing a dataset which overcomes the problem of noisy and uncertain labels by having the authors provide the labels themselves. The shared task is divided into three subtasks:

- SubTask A(English Arabic): Given a text, determine whether it is sarcastic or non-sarcastic;

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 907 - 911

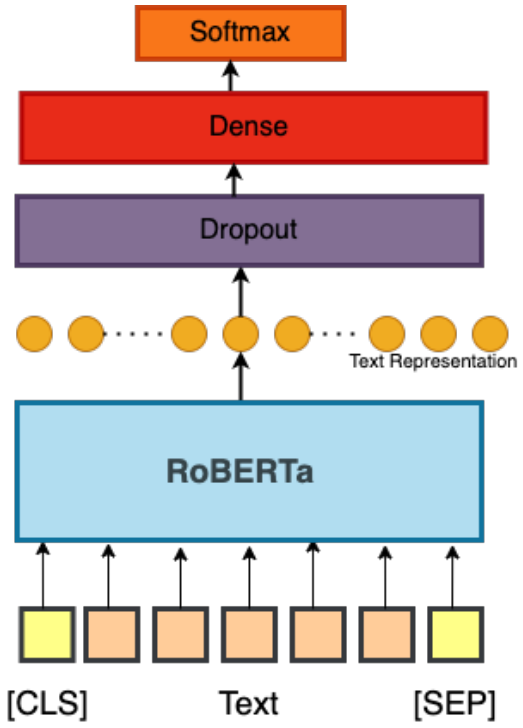

Figure 1: Model Architecture for Subtask A

- SubTask B (English only): A binary multi-label classification task. Given a text, determine which ironic speech category it belongs to, if any;

- SubTask C(English Arabic): Given a sarcastic text and its non-sarcastic rephrase, i.e. two texts that convey the same meaning, determine which is the sarcastic one.

2 System Overview

Here, we describe the data pre-processing steps along with the models for each subtask, and the experimental setup for the system. We provide an overview of the system in Figure 1.

2.1 Data

Manually examining the provided dataset, it was found that the datapoints were mostly from social media. Hence, basic text denoising steps such as user-handle removal, number removal, de-emojifying and repeating punctuation were handled for all three subtasks.

2.2 SubTask A

For this task the participating teams had to develop a system which, given a text, determines whether it is sarcastic or non-sarcastic. We develop a binary sequence classifier as shown in Figure 1. It comprises of a roberta base model as the text encoder with weights taken from cardiffnlp/twitter-roberta-base-sentiment(Barbieri et al., 2020) for English and AraBERT (Antoun et al., 2020) for Arabic. Both of these models are hosted on the HuggingFace library [1] .

The detailed steps involved in this experiment is present below.

- The pre-processed data comprising of cleaned text sequences is tokenized using a BertTokenizer from Huggingface and is passed through the model mentioned above to embed it into a 768 dimensional feature vector containing the syntactical information of the input string.

- The feature vector is then passed through a dropout layer to increase the regularization which in-turn increases the generalizability of the model.

- The model was trained in a supervised manner in a binary classification regime for 5 Epochs with a batch size of 32. Rest of the Hyperparameters are shown in Table 1. A seed value of 42 to keep the model deterministic.

- The model took approximately 45 minutes to train on Nvidia's P100 GPU with a memory of 16Gb.

- The complete experiment was done on Google Colab Pro.

2.3 SubTask B

For this subtask the participants were required to determine which ironic speech category the given input text belongs to. We treated this problem as a multilabel classification with the same encoder as subtask A but with a multilabel classifier instead of a binary classifier to capture the dependency of one sarcasm type on other. We trained this model using a Label Ranking average precision loss for 3 Epochs, with a batch size of 4 and rest of the parameters same as Subtask A for English. Simple Transformers[3] was used to do the development. The model was trained on Google colab and it took

[1] https://huggingface.co/

[3] https://github.com/ThilinaRajapakse/simpletransformers

Parameter	Value
Max sequence Length	96
Batch Size	16
Learning rate	2e-5
Weight decay	Linear
Momentum	0.9
Optimizer	AdamW [2]
Epochs	5
Loss	Cross Entropy

Parameter	Value
Max sequence Length	64
Batch Size	16
Learning rate	2e-5
Weight decay	Linear
Momentum	0.9
Optimizer	AdamW
Epochs	5
Loss	Cross Entropy

Table 1: Experimental Setup for Subtask A

Parameter	Value
Max sequence Length	96
Batch Size	4
Learning rate	1e-5
Epochs	3
Loss	LRAP [4]

Table 2: Experimental Setup for Subtask B

around 35 minutes to finish the training. The Experiment setup has been shown in Table 2 in a concise manner.

3 Results

3.1 Subtask A

All the submitted systems for Subtask A were evaluated using the five metrics that are, Accuracy, Precision, Recall, F1 score, and F1 score of the sarcastic class. However, the ranking of the systems was determined using the F1 score of the sarcastic class. We were officially ranked 10th in Subtask A - English and 11th in Subtask A - Arabic. With an F1 score of 40.16% for English and 40% for Arabic. The rest of the metrics are shown in Table 3.

3.2 Subtask B

The submitted systems for Subtask B were evaluated using the F1 scores of the respective classes in the dataset. The ranking was determined by the Macro F1 score. We ranked at position 22 with a Macro F1 score of 0.0380. The rest of the scores are present in Table 3.

4 Error Analysis

After examining the predictions from the submitted model, we saw that the model struggled significantly in classifying the text sequences to sarcastic type. We inferred that, even though we put less weight on the non-sarcastic class during the loss computation, the model overfitted to the abundant class of non-sarcastic text sequences with a ratio of roughly around 3:1 between the two classes for both Engligh and Arabic tasks. We also noted that individual hyper-parameters had significant roles in the performance of the model. Training different models with different hyper-parameters and ensembling them together showed a significant increase in performance in the post evaluation period. For Subtask B, the main reason for the poor performance of our model and most submitted models on the leaderboard, is the inter-class difference between individual sarcasm types is very low.Which in-turn confuses the model and the output probability is roughly close to each other.

5 Conclusion

We developed a system to classify sarcastic text from non-sarcastic text using contextualized embeddings from a language model which didn't have any prior information about what the fundamental concepts of sarcasm. It inculcates language understanding through self-supervised training techniques namely, masked words prediction and next sentence prediction. However, sarcasm doesn't work in the same way as declarative, exclamatory, imperative, and interrogatory sentences. These were the major type of sentences used for pretraining the language models.

In future work, we plan to use a seq2seq based encoder-decoder model for instilling knowledge of sarcasm in the already available seq2seq models like T5 (Raffel et al., 2019). Wherein we'll use data similar to what was provided in SubTask C and train a seq2seq model with input as the sarcastic text and the model will learn to paraphrase that input sentence into a non-sarcastic sentence as output.

Language	F-1 score	Precision	Recall	Accuracy
English	0.6353	0.6215	0.6683	0.7850
Arabic	0.5800	0.6083	0.7167	0.6571

Table 3: Other Metrics for Subtask A

References

Ibrahim Abu Farha, Silviu Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Tanvi Dadu and Kartikey Pant. 2020. Sarcasm detection using context separators in online discourse. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 51–55.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Elena Filatova. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 392–398, Istanbul, Turkey. European Language Resources Association (ELRA).

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.

Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. 2016. Are word embedding-based features useful for sarcasm detection? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1006–1011, Austin, Texas. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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 preprint arXiv:1907.11692*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Rolandos Alexandros Potamias, Georgios Siolas, and Andreas-Georgios Stafylopatis. 2020. A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, 32(23):17309–17320.

Tomás Ptácek, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on czech and english twitter. In *COLING*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data Knowledge Engineering*, 74:1–12. Applications of Natural Language to Information Systems.

Hamed Yaghoobian, Hamid R Arabnia, and Khaled Rasheed. 2021. Sarcasm detection: A comparative study. *arXiv preprint arXiv:2107.02276*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

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, San Diego, California. Association for Computational Linguistics.