Twitter Sarcasm Detection Using BERTweet

Amy Tzu-Yu Chen
University of Washington
amy17519@uw.edu

David Roesler
University of Washington
droesl@uw.edu

Diana Baumgartner-Zhang
University of Washington
diazhang@uw.edu

Juliana McCausland
University of Washington
jumc1469@uw.edu

Abstract

Sarcasm detection is a task that has grown in importance over the last few years due to the negative impact that sarcasm has on sentiment classification systems Liu, 2010. Sarcasm is inherently difficult to identify because of its form. Here we propose a method to classify sarcasm in tweet text as part of the iSarcasmEval 2022 task. For our approach, we use a BERTweet encoder. Our model outperforms the top scoring submissions to the iSarcasmEval task.

1 Introduction

In any text format, detecting whether someone is sarcastic or serious can be challenging. For spoken interactions, we can rely on intonation, facial expressions, and other non-verbal cues to inform us on the underlying meanings. On Twitter or other social media platforms, users often express their opinions or thoughts solely through text. Discerning whether a text should be interpreted literally or figuratively can prove to be a challenge for humans, and even more so for a machine.

The use of sarcasm on the internet is ubiquitous, and its presence can disrupt computational systems of sentiment analysis, which are widely used in industry Liu, 2010. As a result, it has become essential to build systems that can detect sarcasm. Previous work in this area relies on distant supervision methods that use features like hashtags to indicate whether or not a text is sarcastic. This can lead to noisy data that inevitably produces false positives. Other commonly used approaches rely on the manual labeling of text data. This often requires third-party annotators and can result in problems of annotator agreement.

2 Task Description

Our primary task is sarcasm detection in English Twitter text, which we treat as a binary classification problem. Given a text, our task is to determine whether it is sarcastic or non-sarcastic. Sarcasm is a form of verbal irony through which a speaker expresses their stance toward a topic, which often takes the form of contempt or derogation Wilson, 2006. Automatic sarcasm detection Joshi et al., 2017 is the prediction of the presence of sarcasm in text. Twitter, a platform often used to express the critical viewpoints of its users, has been a common data source for sarcasm detection models Sarsam et al., 2020. To train and evaluate our model, we make use of the Twitter sarcasm dataset from SemEval 2022 Task 6, iSarcasmEval Oprea and Magdy, 2020. Unlike sarcasm datasets labeled by third-party annotators, this dataset contains labels provided by the authors of the tweets themselves. The iSarcasmEval data includes both English and Arabic sets.

As our secondary adaptation task, we perform sentiment detection on the Arabic portion of the dataset. To evaluate the performance of our binary classification model, we measure F1 score on the positive (sarcastic) class.

3 System Overview

To perform sarcasm detection, we fine-tune BERTweet Nguyen et al., 2020 on the iSarcasmEval training data. As illustrated in Figure X, we attach a classifier head (a pair of fully-connected layers with a softmax output) to the BERTweet base model. The classifier head produces a sarcasm prediction for each input tweet using the [CLS] token representation produced by the BERTweet encoder.

4 Approach

The tweets included in iSarcasmEval dataset exhibit typical characteristics of Twitter text, which is short in length (due to character limits) and contains frequent use of informal grammar and irregular vocabulary, such as abbreviations and hashtags (Han et al., 2013). Additionally, the iSarcasmEval data is rich in emoticons and emoji, both of which

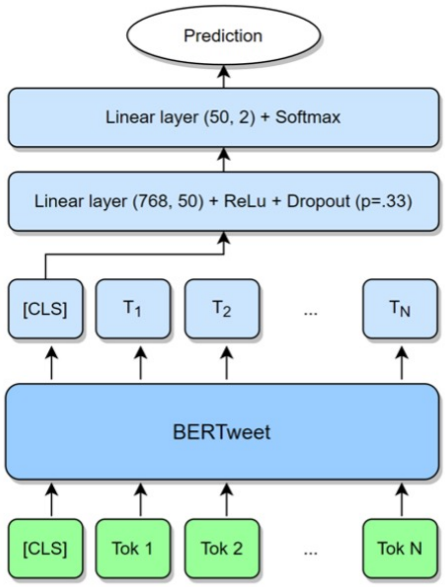

Figure 1: Fig. X: Overview of classification model

are used by social media users as nonverbal cues to indicate sarcastic intent (Filik et al., 2016).

In order to leverage the unique grammatical, lexical, and symbolic characteristics of tweets in our classification system, we utilize BERTweet, a large-scale language model trained on 850M tweets, as our encoder. To preserve emoji information in inputs, we first normalize tweets using the tweet normalizer function used in the BERTweet pre-training by Nguyen et al. The tweet normalizer (Fig. X.2) converts emoji into text strings, user mentions into '@USER', and web links into 'HTTPURL'. We bal-

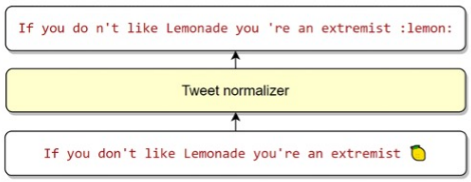

Figure 2: Fig. X.2: Example tweet normalization

ance the number of positive and negative examples of sarcasm in our dataset by including all examples of the positive class and randomly sampling an equal number from the negative class. For fine-tuning, we use a single Tesla P100 GPU on the Google Colab platform. Pytorch and the Huggingface library are used to fine-tune BERTweet on the iSarcasmEval training set for five training epochs. We use AdamW (Loshchilov Hutter, 2017) with a learning rate of 5.e-5 (controlled by a learning rate scheduler) and a batch size of 32.

As our baselines, we use Random Forest and LightGBM to train binary classification models for predicting sarcasm or not. In the preprocessing step, we used TfidfVectorizer with n-gram range set to (2, 6) so we consider bi-gram to six-grams. After vectorization, we run Random Forest and LightGBM using the same balanced training data set.

5 Results

Model	Accuracy	Precision	Recall	F1	Macro F1
Random forest	.53	.53	.66	.59	0.53
LightGBM	.53	.52	**.76**	.62	0.50
BERT-base (cased)	.58	.57	.58	.58	.58
RoBERTa-base	.69	.69	.67	.68	.69
BERTweet	**.73**	**.76**	.69	**.72**	**.73**
iSarcasmEval top F1 (test set)				.61	

Figure 3: Table X: Validation results for positive (sarcastic) class

The baseline models performed well when compared to the official iSarcasmEval results, which utilize F1 scores. The top three F1 scores on the official iSarcasmEval results list are .61, .57, and .53, meaning our Random Forest baseline performed better than the second place participants, and the LightGBM baseline model performed better than the first place participants. When we decided to use a language model, we initially chose BERT. This model achieved a lower F1 score than the LightGBM and RandomForest baselines, although its Macro F1 saw an improvement. We then tried RoBERTa and saw a noticeable improvement across the board. We realized that the linguistic properties of tweets may call for a language model that was specifically trained on tweet data. This was when we decided to go with BERTweet. Our improved BERTweet model performed best, with a .72 F1 score, with improved accuracy, and precision scores compared to the other models. Our BERTweet model did, however, achieve a lower recall score than the LightGBM model.

6 Discussion

Our first obstacle was in developing an approach. While our LightGBM and RandomForest baselines performed well compared to the other participants' models for the iSarcasmEval task, we felt it necessary to attempt to use a pre-trained language model. In doing so, we were able to observe the importance of selecting the right language model for a

given task. On the other hand, we found it strange that even our baselines achieved such high performance compared to the other submissions for this task.

We analyzed the data more deeply and found that there were various annotations we disagreed with. Tables X.2 and X.3 below contain two examples each for positive class annotations and negative class annotations with which we disagree.

Original Tweet	Rephrase (from tweet author)	Labels
"You like ketchup but not tomatoes???"	It seems childish to enjoy ketchup but not tomatoes	sarcasm
"CLB cover art makes me want to pound my knees with a hammer"	I do not like the certified loverboy artwork	overstatement

Figure 4: Table X.2: Tweets labeled as sarcastic in original dataset

Original Tweet	Our interpretation
"Oh cool i drive to school in less than 12 hours and I'm frantically looking up fun facts about the final destination franchise for absolutely no reason it's fine everything's fiNE"	Everything is not fine. The author is anxious after reading about final destination right before having to drive long distance
"Ah yes. After a decent birthday party it is now time to stress out about money again."	The 'ah yes' carries a connotation of pleasantness, making this statement seem ironic/sarcastic

Figure 5: Table X.3: Tweets not labeled as sarcastic in original dataset

The first tweet in Table X.2 says "you like ketchup but not tomatoes?", which is then rephrased by the author as "it seems childish to enjoy ketchup but not tomatoes". The author labeled this as sarcasm, however, we do not recognize this as such. Another tweet that was annotated as sarcastic says "CLB cover art makes me want to pound my knees with a hammer", which was rephrased by the author as "i do not like the certified loverboy artwork". This also does not fit the definition of sarcasm, since the surface level message of the original tweet matches the underlying meaning and the rephrasing. It is accurately labeled as an overstatement, but, to us, this does not constitute sarcasm.

Following the same line of logic, the tweets in Table X.3 fit the criteria for sarcasm by displaying a positive surface level sentiment, with a negative underlying meaning. They are both executed in a way that we believe to be sarcasm.

While we initially felt that having the dataset annotated by the authors of the tweets themselves would lead to higher agreement and consistency within the dataset, we realized that this actually may have led to less consistency and agreement in the data, which may invalidate the performance of models training on this dataset. It may be worthwhile to create a more narrow definition of sarcasm in future work using this dataset.

In future iterations of our model, we plan to combine the BERTweet [CLS] output with additional representations of the tweet and use the combined representations as the input to the classifier head. One possible source of additional representations is a DeepMoji encoder. DeepMoji (Felbo, et al., 2017) is a model that was trained to predict the emojis correspond to particular tweets. It was premised on the idea that if the model can accurately predict the emojis for a given text, then it must also grasp, to some extent, the sentiment of that text. DeepMoji has been explicitly used for sarcasm (Felbo, et al., 2017) (Walker, et. al, 2012), and could give interesting results when combined with our current BERTweet representations.

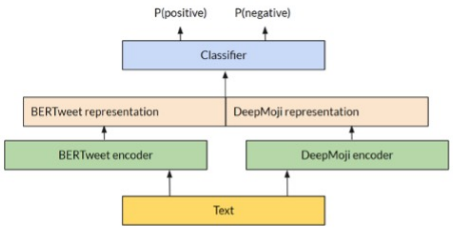

Figure 6: Fig. X.3: Combining multiple representations

7 Conclusion

[DRAFT]Our BERTweet model outperformed the highest scoring submissions to the iSarcasmEval task, with an F1 of .72.

References

Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: A survey. *ACM Comput. Surv.*, 50(5).

Bing Liu. 2010. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing*.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Silviu Oprea and Walid Magdy. 2020. iSarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1279–1289, Online. Association for Computational Linguistics.

Samer Muthana Sarsam, Hosam Al-Samarraie, Ahmed Ibrahim Alzahrani, and Bianca Wright. 2020. Sarcasm detection using machine learning algorithms in twitter: A systematic review. *International Journal of Market Research*, 62(5):578–598.

Deirdre Wilson. 2006. The pragmatics of verbal irony: Echo or pretence? *Lingua*, 116(10):1722–1743. Language in Mind: A Tribute to Neil Smith on the Occasion of his Retirement.