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

3.1 Initial System

In our initial sarcasm detection model, we fine-tune BERTweet Nguyen et al., 2020 on the iSarcasmEval training data. As illustrated in 1, 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.

3.2 Revised System

In our revised sarcasm detection system, we change the encoder component of our system from a BERTweet-base to a BERTweet-large model. The BERTweet-large model has the same structure as RoBERTa-large Liu et al., 2019, which produces 1024-dimensional output representations, rather

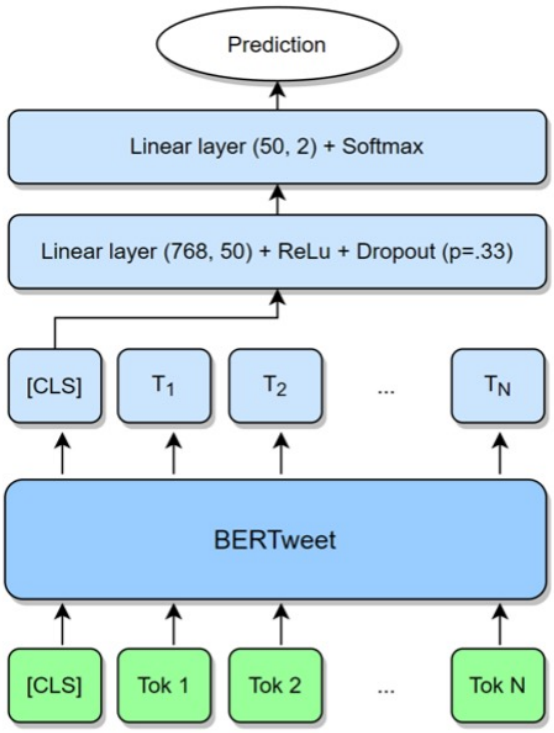

Figure 1: Overview of classification model

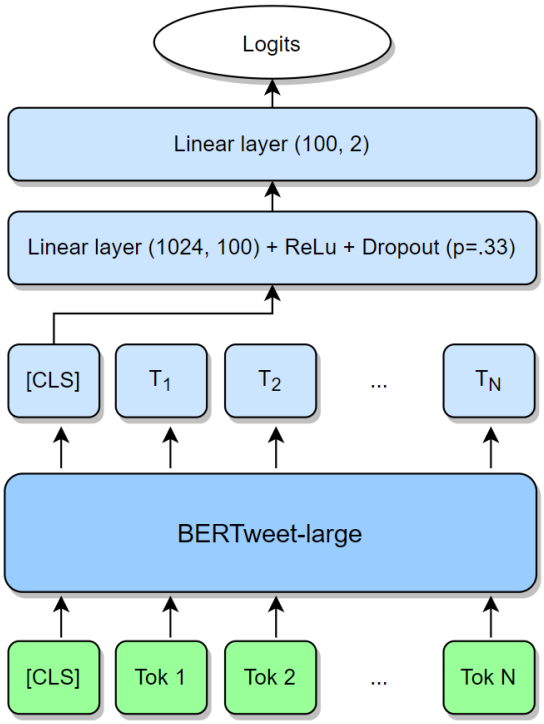

Figure 2: Overview of revised BERTweet-large classifier

than the 768-dimensional outputs produced by the base version of the model. As shown in 2, we modify our classifier head to match the output size of BERT-large and also increase the size of the hidden layer from 50 to 100 neurons.

To create our system's sarcasm predictions, we create an ensemble of five BERTweet-large models, depicted in 3. Each of the models in the ensemble are fine-tuned on our original training set augmented with a different random sampling of tweets from the English sarcastic tweet dataset produced by Ptácek et al., 2014. Our ensembling strategy is variance reduction through bootstrap aggregation, or bagging, where hard majority voting is used to create the final system predictions.

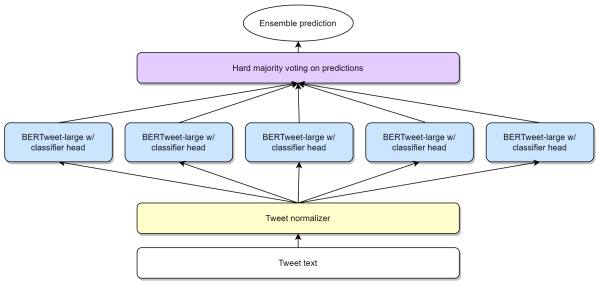

Figure 3: Overview of ensemble model

4 Approach

4.1 Initial 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. Additionally, the iSarcasmEval data is rich in emoticons and emoji, both of which 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 (4) converts emoji into text strings, user mentions into '@USER', and web links into 'HTTPURL'.

We balance 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 Hug-

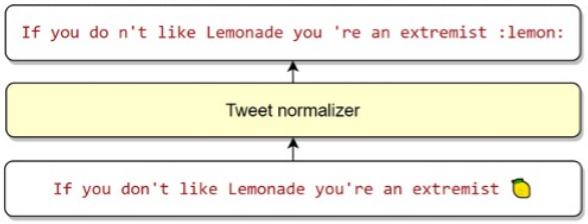

Figure 4: Example tweet normalization

gingface library are used to fine-tune BERTweet on the iSarcasmEval training set for five training epochs. We use AdamW(Loshchilov and 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.

4.2 Revised Approach

4.2.1 Submitted System

In our revised approach we create sarcasm predictions for each tweet using an ensemble of BERTweet-large classifiers. Each of the five classifiers is trained on our original balanced training set created from the iSarcasmEval training data Oprea and Magdy, 2020. Additionally, the training set for each of the classifiers is augmented with a random sampling of 800 positive and 800 negative tweets from the Twitter sarcasm dataset produced by Ptácek et al., 2014.

The Ptácek et al., 2014 data differs from the iSarcasmEval data in that sarcastic tweets were identified through distant supervision (hashtags such as sarcasm) rather than author identification. Additionally, the Ptácek et al., 2014 data is roughly six years older than the iSarcasm set and does not contain more recent lexical innovations, such as terms related to COVID-19 and covid vaccination.

We fine-tune each of the sub-models on a single Tesla P100 GPU on the Google Colab platform using the AdamW optimizer with a learning rate of 3e-6 (controlled by learning rate scheduler) and a batch size of 32. We train for 30 epochs and retain the checkpoint where the model achieves the highest F1 score on the positive (sarcastic) class. When fine-tuning each of our ensemble sub-models, we also utilize thresholding Sheng and Ling, 2006 to identify the optimal threshold for best positive class F1 for each submodel.

The logits output of each sub-model are passed through a softmax function and predictions for each sub-model are determined using the best probability threshold for that model (as determined during training). The predictions for each sub-model are aggregated and the final ensemble predictions are determined by hard voting.

4.2.2 Experiments

In our system revision process, we also experimented with a few changes that did not enter our final submitted system.

We combined the BERTweet [CLS] output with fine-tuned DeepMoji Felbo et al., 2017 representations of the tweet and used the combined representations as the input to the classifier head. This method provided minimal improvements when combined with the BERTweet-base model and no improvement when combined with BERTweet-large, and was not included as a component of our final system.

To supplement our data, we attempted the addition of the Self-Annotated Reddit Corpus (SARC) Khodak et al., 2018. Ultimately, we did not use this dataset in our revised system. Since the length of Reddit posts can be much longer than tweets, and also due to the absence of emojis and hashtags, we felt the dataset was not suitable to the task and current system.

5 Results

5.1 Initial 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 5: 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.

5.2 Revised 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-base	.74	.75	.72	.73	.74
BERTweet-large	.80	**.81**	.79	.80	.80
BERTweet-large w/ data augmentation	.80	.77	**.86**	.81	.80
Ensemble w/ data augmentation	**.81**	.78	**.86**	**.82**	**.81**
iSarcasmEval top F1 (test set)				.61	

Figure 6: Revised system validation results for positive (sarcastic) class

In our revised system, we used BERTweet-large and saw an improvement with an F1 score of .80, an increase of .07 compared to the BERTweet-base. With data augmentation to the BERTweet-large system, we saw a slight gain with an F1 of .81. The five classifier BERTweet-large ensemble (with data augmentation) performed the best out of our experiments, with an F1 of .82.

6 Discussion

6.1 Challenges and Limitations

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. Since we did not evaluate our system on the iSarcasmEval's official test set, it is possible the task's test set is more difficult than our development dataset.

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. In our error analysis, we highlight some of the issues in iSarcasmEval with specific examples.

6.2 Error Analysis

In our revised system we observed a few patterns in prediction errors. Some of the false negatives require more context, world knowledge, or information about the author in order to determine the presence of sarcasm, as shown in 7. The first tweet requires knowledge of "queen's gambit" as well as the author's underlying opinions on the show. Similarly, the second tweet requires both knowledge of the author's opinions and the show "degrassi".

Original Tweet
"finished the queen's gambit, i loved a look into the 1960s in the american south with no racism just vibes"
"rewatching degrassi for the millionth time"

Figure 7: False negative predictions that required additional contextual information

Also, some of the false positives appeared to be mislabeled. In 8, we highlight two such examples. In both, we believe that the intended meaning fits the definition of sarcasm, although the tweet is not labeled as such.

Original Tweet	Our Interpretation
"People who drive under the speed limit. Death Penalty."	People who drive under the speed limit annoy me.
"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

Figure 8: Tweets not labeled as sarcastic in original dataset that we believe to be sarcastic

Additionally, there are some false positive examples that lead us to believe the model is learning spurious patterns. In 9, we highlight two tweets in which neither appear to have sarcastic intent, but were predicted to be sarcastic by our model.

Original Tweet
“hey twitter, how do you twitter now a days.”
“i love how quiet it gets when it snows”

Figure 9: False positive examples with no clear sarcastic intent

6.3 Future Work

With future work on the English portion of the iSarcasmEval task, it may be worthwhile to create a more narrow definition of sarcasm, and remove confusing or mislabeled examples in the iSarcasamEval dataset. Additionally, updating the annotations and labels on the tweets may be helpful as well.

Also, we will attempt to use our system on the Arabic iSarcasmEval dataset. We hope to perform the same sarcasm detection task on the Arabic portion of the dataset, and evaluate our performance by measuring the F1 score on the positive (sarcastic) class.

7 Conclusion

In our initial system, the BERTweet model outperformed the highest scoring submissions to the iSarcasmEval task, with an F1 of .72.

With our revised system, we used an ensemble of BERTweet-large classifiers,and included an additional dataset from Ptácek et al., 2014. Our ensemble system saw an improvement of .11 over our initial one, with an F1 of .82.

References

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.

Ruth Filik, Alexandra »öurcan, Dominic Thompson, Nicole Harvey, Harriet Davies, and Amelia Turner. 2016. Sarcasm and emoticons: Comprehension and emotional impact. *Quarterly Journal of Experimental Psychology*, 69(11):2130–2146. PMID: 26513274.

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

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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

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.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

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.

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

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.

Victor S. Sheng and Charles X. Ling. 2006. Thresholding for making classifiers cost-sensitive. In *Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference, AAAI-06/IAAI-06*, Proceedings of the National Conference on Artificial Intelligence, pages 476–481. Null ; Conference date: 16-07-2006 Through 20-07-2006.

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.