A Variation of Feature-based Machine Learning Approach for Recognizing Sarcasm

Luheng (Wes) Wang
New York University
lw2534@nyu.edu

Haoyuan (Kyle) Ma
New York University
hm1920@nyu.edu

Ruixuan (Rachel) Xing
New York University
rx493@nyu.edu

Abstract

Sarcasm classification has been a popular research area, and previous approaches were mainly two types: training machine learning models with feature extraction and fine-tuning a pre-trained language model such as BERT (Devlin et al., 2019). However, the features used were mainly syntactic tagging. We argue that syntax, semantics, sentiment, and POS tagging can all be helpful in producing a better performance in classification. Thus, we propose three different methods based on this hypothesis. First, We design machine learning systems based on features of these tags. Second, we fine-tuned a BERT-base model with regular embeddings as input. Third, we combine the results of BERT and the features we designed. We anticipate the output of BERT to contain more semantic information about the text and complement those explicit tags. We focus on data from the domain of Reddit, in which text is in the format of micro-blogs. Our results show that generally BERT itself outperforms regular feature-based machine learning models as well as the combination of BERT's outputs and our features.

1 Introduction

Sarcastic texts are hard to classify, even for humans. It requires clear context to determine whether a piece of text is sarcastic, and it is often obscure if only provided with a single sentence. In other words, the relationships between and within sentences (inter-/intra-sentence) are essential to understanding these sentences. Therefore, it is logical to explicitly teach the models these relationships to make valid classifications. The challenge is defining these relationship and designing features which represent them. Many researchers have conducted extensive researches in the domain of social media such as Twitter. Nevertheless, most probing research investigated sarcasm recognition using only a series of surface patterns. For instance, one strategy was to identify sarcasm by exploring user comments for oral and gestural clues such as expressions, laughter, and onomatopoeia (Carvalho et al., 2009). However, the authors of this strategy themselves also indicated in this paper that only using these surface patterns is not sufficient to achieve a good performance. Moreover, some early research tended to eliminate the influence of emoji and punctuation (Lichouri et al., 2021). We argue that these are all important subtleties that might contribute significantly to the sarcastic side of a piece of text, and they will help the systems to classify sarcasm. Intuitively, we humans also obtain crucial information from emoji and punctuation upon deciding if some text is sarcastic. Therefore, we include them in our training on Reddit comment data.

Results from Ptácek et al. (2014) clearly show that machine learning systems based solely on punctuation or general patterns, such as word frequency, will be outperformed by systems that have some knowledge about the sentence structure, such as the ones with POS tagging implemented. These results facilitate our motivation of incorporating similar features in our study. We use POS tagging and sentiment tagging to examine inter-/intra-sentence relationships. We utilize the VADER Sentiment Analysis tool (Hutto and Gilbert, 2014) for this purpose, which is specifically tuned for social media text. It works perfectly on data in the format of microblogs, and it can systematically quantify the relationship in sentences based on POS tagging. In addition, we adopt the design of *written-spoken* feature in the work of Barbieri et al. (2014a) based on ANC corpora, but we simplify the feature to be purely binary. To sum up, we utilize some manually-designed features for the feature-based method, which focuses on sentence relationships. We borrow a few features from past research such as *written-spoken*, and we also extract the tradi-

tional features in sentence classification tasks, such as *punctuation*.

Next, we fine-tune a BERT-base model on the same data. The BERT-base model is an independent experiment different from the previous feature-based machine learning approach. The past research suggested that BERT can learn syntactic as well as structural information in text (Jawahar et al., 2019) on its own. Thus, we believe that although we designed an array of features specifically for sarcasm recognition, BERT can acquire this knowledge on its own. We hypothesize that BERT, as a pre-trained neural network, will outperform our machine learning approach, and we can study how much it outperforms the machine learning method.

Lastly, we design a way to combine the BERT results and our features. Despite the claim that BERT can learn these features itself, and therefore the results should be based on this information, we investigate whether it helps to emphasize these features explicitly. We achieve this experiment by inputting the outputs of BERT, i.e., the logits for classification, to the machine learning models we use. Basically, we view the outputs as yet another feature. We innovatively design this method, but we can only infer its validity and effect through empirical results. According to the final output, the combination of BERT and feature-based machine learning performs slightly more poorly than the method of using only BERT, but outperforms the pure feature-based machine learning method. However, we suggest that the improvement is solely due to the goodness in BERT's output. In all, we should trust the capability of BERT to learn all features implicitly. In addition, pre-trained language models transcend the traditional machine learning models significantly.

2 Related Work

Some of the early work on the task of sarcasm recognition focused mainly on lexical and syntactic features. Kreuz and Caucci (2007) did a comprehensive study on the part of speech tags and punctuation and showed the importance of them upon recognizing sarcastic linguistic expressions. They also focused on specific tokens such as "gee" and "gosh", which inspired us that certain words might be more definitive in the context of sarcasm, and we have developed some of our features based on this idea. Some other early works also emphasized the importance of contradicting sentiment values in an expression, such as "love" followed by some unpleasant object or experience. In the research on how sarcasm interacts with contrasting sentiment by Riloff et al. (2013), the author illustrated that these sentiment values can be a good factor in classifying sarcasm. However, as claimed by the author, one potential improvement of their research was to capture the syntactic structure of the input sentence(s). For instance, their classifier often misses the prepositional phrases attached to contrasting tokens. As one of the examples in their paper, "I love fighting with the one I love" was correctly classified as sarcastic, but their system made the decision based on the contrasting tokens "love" and "fighting." However, the token "fighting" does not contain enough context for accurate classification; instead, the prepositional phrase "with the one I love" should be the deterministic factor here. Therefore, though correctly classified sarcasm, the system built this correct classification on a flawed basis. The way to improve is to add more explicit boundaries on syntax and structure, which we fulfill with POS tagging.

Later work often incorporated at least some syntactic structure representations in the classifiers. For example, the use of n-gram and POS tagging in the work of Ptácek et al. (2014) showed substantial improvement over their baselines. Moreover, studies done in the domain of social media, such as twitter, provided a solid foundation for our feature-based method. Other than some inspiring features they designed, we also can understand the difficulty with the online text. For instance, the work by González-Ibáñez et al. (2011) clearly states that Twitter text is hard to classify due to its common short length and lack of context. Our selection Reddit data is in a similar format, micro-blogs, and has similar features, such as replying to other users. Therefore, we recognize and discuss these obstacles in our error analysis.

Furthermore, since the current state-of-the-art language models are pre-trained on a large corpus of text, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021), we run experiments on them to see if the language information captured during pre-training can make up for the lack of context in sarcasm recognition tasks. In addition, according to the authors of these models, the information of the features we extract should be learned by these models in their complex architecture and

high-dimensional word embeddings.

3 Methodology

3.1 Baseline

There is no published research studying the same dataset we chose; thus, we design our own baseline. Since our data is perfectly balanced, we can use a trivial baseline, i.e., 50% accuracy (predicting every record to be sarcastic). We compare each of our three methods of detecting sarcasm to this trivial baseline.

3.2 Feature Extraction

We extract two types of features. The first type of features we focus on is a combination of computational semantics and POS tagging, which includes:

- Intra-sentence contradiction: We examine if there are strong oppositions within each sentence in the text. Such contradiction includes both strong positive and negative (contradicting) adjectives or adverbs, existences of contradicting nouns in a sentence, existences of contradicting adjectives and nouns in a single Noun Phrase (NP), and existences of contradicting adverbs and adjectives in an ADJP or NP. We can quantify this feature by assigning values to words based on positivity with VADER Sentiment Analysis tool (Hutto and Gilbert, 2014).
- Inter-sentence contradiction: We do the same with consecutive sentences as above.
- Negating coordinating conjunction: We examine if there exists CCs with potential negating context within a sentence, i.e., but, or, nor.
- Negating adverb between sentences: We examine if there exists RBs with potential negating context between sentences, i.e., However, Nevertheless.

The second type of features we extract focus on special characteristics of sentences which we believe are good indicator of sarcasms:

- All-cap word: We inspect whether there is all-capitalized word in the text.
- Count of punctuation: We count the appearances of punctuation per sentence.
- Written-spoken: We evaluate whether the text is of spoken or written style based on the work by Barbieri et al. (2014b).

3.3 Method 1: Feature-based Machine Learning

Our first step is to use a combination of feature extraction and machine learning to detect sarcasm in our Reddit dataset. The pipeline for this method is shown in Figure 1. As discussed in Section 1, we adopt the research outcome from previous researchers and design our features to incorporate: 1) formal sentence structure using POS tagging for inter/intra-sentence analysis and 2) indicators of strong emotions such as punctuation and all-capitalized words. We aim to design features from the following aspects of a linguistic expression: syntax, sentence hierarchical structure, semantics, and styles.

We first constructed a word list based on the tokenized Reddit data to help determine the feature marks. Then we built the input features list with the first part as the input sentence(s) in the data set and the second part as our feature marks for the input sentence(s). Most feature marks should be either 0 or 1, indicating whether the input data possess the designed feature or not, except for the count of punctuation feature that is an integer, showing the number of punctuation included in the input sentence(s). Afterward, we feed the features into a logistic regression model. We train it for one epoch and use the trained model to predict the testing set and obtain scores for multiple metrics, which we discuss in Section 6. We repeat the training for 5 trials with different ordering of our dataset to ensure the results we obtain are not due to random variation from initializing the model or the training process. Moreover, we utilize a second decision tree model to generalize our results. Thus, our results should be not model-specific, and they are mainly due to the features instead of the model.

We follow the traditional machine learning procedure, with 5-fold cross validation implemented. However, due to a limited number of features, we suspect that the model might not learn the input text comprehensively. Limited features indicate limited dimensions from which the models can acquire information from the input. Therefore, the models might not be able to produce reasonable classifications based on the constrained context learned from the features.

BERT (Devlin et al., 2019), on the other hand, learns semantics on token-level instead of sentence level. We feed each token to the model as a 768-dimensional embedding. The dimension of infor-

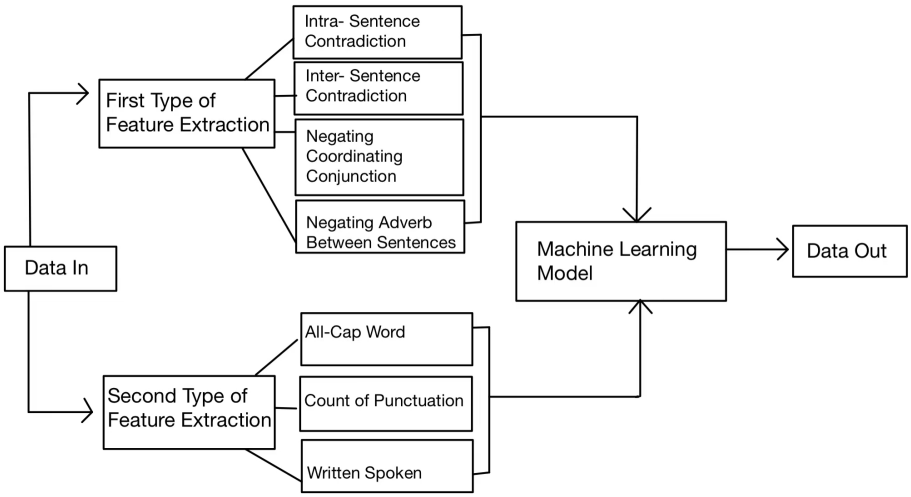

Figure 1: The pipeline of Method 1

mation transcends our feature-based input. Moreover, as an enormous pre-trained language model, it already possesses abundant language information, and it is capable of learning all features we defined implicitly in its 12-layer architecture. Therefore, we hypothesize that we would have a stronger classifier by fine-tuning a BERT model.

3.4 Method 2: BERT

As discussed before, we utilize BERT-base to fine-tune our sarcasm Reddit dataset. We used BERT-base for shorter training time, specifically the *BertForSequenceClassification* configuration since our study is a binary classification task. We train the model using GPUs on NYU Greene cluster. We follow the regular steps discussed in the BERT paper to input our data. First, we tokenize each word. Then, we feed them into the embedding layer as well as the rest of the model. After each epoch, we validate on our validation set. Finally, when we finish all epochs, we predict the testing set. We repeat the training for 5 trials to ensure the results we obtain are not due to random variation from initializing the model or the training process.

Since our dataset contains a hundred thousand records, the output of BERT is a 100000×2 array, in which each pair of two numbers represents the two logits BERT calculated for either sarcastic or non-sarcastic. We will convert this array to an additional feature in the next method. For this method, we choose the label represented by the larger logit and classify this data entry to be that label. Table 3 shows three sample outputs of BERT. In this example, we classify the first and third records to be sarcastic and the second one to be non-sarcastic.

Logits Non-Sarcasm	Logits Sarcasm
-2.6014194	2.762062
0.402185	-0.31562826
-2.5029473	2.5489013

Table 1: Sample output of BERT

3.5 Method 3: BERT with Feature-based Machine Learning

Our third approach to detect sarcasm is to mix BERT's output with our feature-based machine learning models. We utilize the Non-Sarcasm and Sarcasm Logits produced by BERT as two additional features. The dataset processed and marked by BERT is split into 80% training and 20% testing datasets. The BERT logits, along with other features mentioned and discussed in Section 3.3 and 3.2, are used to train a logistic regression model and a decision tree model. We select the same two models in order to compare results with our feature-based machine learning method discussed in 3.3. We repeat the training for 5 trials with different ordering of our dataset to ensure the results we obtain are not due to random variation from initializing the model or the training process. The max and average accuracy, precision, recall, and F1-score over the five times are calculated and recorded.

4 Data

Our dataset is the "Sarcasm on Reddit" dataset collected by Khodak et al. (2017) for their article "A Large Self-Annotated Corpus for Sarcasm" (Khodak et al., 2017). The dataset contains 962,296 unique comments from the internet commentary website Reddit. Since Reddit is a social media platform, its data is well compatible with the VADER sentiment analysis tool, which is specifically attuned to social media text. Redditors use the tag to indicate that their comment is sarcastic, which is generally a reliable indicator of sarcastic comment. The dataset contains label (1 indicates the text is sarcastic, 0 indicates non-sarcastic), comment text, author, origin subreddit (topic of comment), comment score, comment upvote, downvote, created date, created datetime and parents comment columns. In our data, we mainly focus on the comment text and sarcasm label column. We split the dataset into a 20 percent testing set and 80 percent training set for further analysis. Table 2 shows a few sample sentences and their corresponding labels.

Comment	Label
Why not both?	0
I'd be creeped out by either drunken sex.	0
I still like eight year old girls.	1
I'm a PoGo player and i'm pretty sure...	1

Table 2: Sample Data

5 Experiment Configuration

We use the exact same configuration for Logistic Regression and Random Forest in Method 1 and 3.

5.1 Logistic Regression

- Penalty term: L2
- Tolerance: 1e-4
- Solver: LBFGS

5.2 Decision Tree

- Criterion: Squared Error
- Splitter: Best
- Min of split: 2
- Max depth: None

5.3 BERT

- Tokenizer: BertTokenizer (recommended)
- Tokenizer maximum length: 128
- Batch size: 32
- Epoch: 4
- Optimizer: Adam
- Learning rate: 2e-5

6 Results and Analysis

Results are listed in Table 3. We first recognize that the feature-based decision tree performs better than feature-based logistic regression. The decision tree model also performs consistently under all 4 metrics, while logistic regression tends to perform poorly when evaluated by recall. This indicates that the logistic regression model mis-classifies many sarcastic expressions. The decision tree appears to be stronger and less biased between these two models. Overall, they both perform much better than our baseline.

BERT, on its own, performs the best among all three methods. It is also stable across different metrics. As discussed, BERT is capable of learning implicit linguistic and semantic features. The results support this claim because BERT can still achieve relatively high performance despite the lack of context of these Reddit posts. This accuracy is comparable to humans because without considering the context in the chain of replies, it is hard for us to classify, too.

Next, the combination of BERT's results and our features in the two models from Method 1 does not improve the overall performance. This result directly discourages the explicit specification of the features. In other words, we should trust the BERT's capability of learning required information (including those features) instead of manually specifying them. Moreover, since the two logits produced by BERT are already the essence of all the knowledge BERT has for the data, it would be unnecessary to input them into less complex models (logistic regression and random forest). Based on our empirical results, this method seems to be helpless. One interesting finding is that when the two models from Method 1 are trained with the designed features, the decision tree performs much better than the logistic regression. However, after feeding in the BERT's results as features, decision tree performs much worse than the logistic regression. The combination of our designed features

Models	Accuracy	Precision	Recall	F1-score
Feature-based Logistic Reg	0.6232	0.6441	0.5498	0.5587
Feature-based Decision Tree	0.6722	0.6441	0.6304	0.6326
BERT	0.7466	0.7601	0.7332	0.7464
Feature-based Logistic Reg + BERT	0.7404	0.7521	0.7277	0.7397
Feature-based Decision Tree + BERT	0.6544	0.6620	0.6503	0.6561

Table 3: Results from three methods. Method 1 and 2 have two models. Each number is the average of 5 trials.

and BERT's results has improved the performance of logistic regression to nearly the level of BERT's performance. However, the decision tree does not seem to benefit from these additional inputs. Our inference of this phenomenon is that tree-based models generally favor categorical data instead of numerical data. Based on the work by Liu et al. (2009), numeric input does not work well with our decision tree model might be because the tree is binary. The information contained in the logits is meaningless to the tree as it has one single decision boundary when splitting. The paper also suggests that if the tree is multi-split, it will be more flexible and hence able to utilize these numeric values and we will leave this for future research.

7 Error Analysis

To analyze the gaps between the three methods, we select several pieces of text and their respective predictions by each method. As shown in Table 4, we pick four typical types of text that each method misclassifies.

No.1 text is misclassified by feature-based method (Method 1 from Section 3.3). This piece of the text shows one weakness of our feature-based method. That is, there will likely be some input text that contains very few of our pre-defined features once in a while. For example, this sentence has no negating CCs (but, nor), no negating RBs (however, nevertheless), no intra-sentence contradiction (contrasting sentiment), and no all-cap words. Therefore, the features help are very limited in classifying this sentence. The strength of BERT can make up for this flaw, as it has already learned tremendous linguistic information during pre-training, which helps it handle unseen text from more angles than these features.

No.2 text is misclassified by BERT (Method 2 from Section 3.4) and the Mixed method (Method 3 from Section 3.4). Note that this is a very long piece of text (118 words). We suggest the reason that BERT and the Mixed method fail is the length. Since the self-attention mechanism in BERT models is designed to detect each token's relationship with others, it may not work well on really long input, especially with a lot of repetitions. For example, in this text, the word "temple" is repeated multiple times, and it is relating to the last several phrases which are the source of sarcasm. However, our feature-based method will not be affected by the length. We roughly calculate the average length (in tokens) of text that is misclassified by Method 2 and 3 and text that is misclassified by Method 1, and it turns out the former is longer than the latter by 20%. This result provides evidence to support our suggestion. On the other hand, since the input is long, more features are captured during extraction, contributing to the classification.

No.3 text is misclassified by all three methods. Based on the sentence alone, it is difficult to make the proper classification even by human annotators. The author of this text created this label, so the author was aware of the context when they wrote this post. However, we do not have access to any context except for what is in this text. This sentence can be either sarcastic or non-sarcastic based on a different context. Therefore, we conclude that the failure to make a correct classification in this text does not provide any insight into how the three methods differ from one another.

Most of the time the classifications of BERT and the Mixed methods are the same, showing the BERT logits are dominating the Mixed method. Thus, the rarest combination is when BERT alone makes the proper classification while the rest two methods do not. We pick one example from this rare scenario, text No.4. There are many features we designed existing in this particular sentence, such as all-cap words ("TOO," "FAR"), negating CCs ("but"), spoken-style ("might have been," "just took it"), etc.. However, the two methods which utilize these features fail while BERT alone succeeds. Our inference of this case is that these features in this example do not help and will even sabotage the

No.	Text	Feature-based	BERT	Mixed	True label
1	Finally, someone is talking about the *real* issues!	0	1	1	1
2	TBM mother goes to temple ... God would cure me of my 37 year case of the gay	1	0	0	1
3	I'm a big fan of wings.	0	0	0	1
4	I might have been interested after the first two but the third just took it TOO FAR.	0	1	0	1

Table 4: 1 represents sarcastic (either predicted or true label), 0 represents non-sarcastic. Incorrect classifications are marked red. Note that No.2 text is truncated since it is too long (120 words).

model's judgment. However, these cases are rare, and we believe they might be due to poor annotation of the original data since sarcasm is a rather subjective linguistic property.

One thing to worth noting is that

To sum up, we have selected a few examples representing the majority in each case (correctly classified by some methods but not the others). We notice that BERT tends not to perform as well when the input is long. In addition, the Feature-based method has a limitation in that the input might not express most features at all. Moreover, even if the features are mostly evident, it does not guarantee that the methods with features as input would perform well. We have made our inferences for each of these observations.

8 Conclusions

We examine three methods of detecting sarcasm in input text: Feature-based, BERT, Mixed. Our findings indicate that BERT by itself is the most powerful method, and there is no need to combine the features with BERT explicitly. We ought to trust its learning capabilities. The feature-based machine learning method also performs well compared to the baseline, especially when text is short and lacks context. Moreover, through our error analysis, we can reasonably infer that adding the features to the input can make an explainable impact on the classification, though not always a positive impact.

It remains an open question whether we can improve BERT's performance by using other methods of combining features with it. Furthermore, it is also unclear if the failure in this combination (Method 3) is due to our feature design. Future work could perform ablations over each feature or potentially design more new features.

Acknowledgments

We sincerely thank Professor Adam Meyers for his guidance on how to conduct our analysis. We sincerely thank Shubham Vatsal for providing advice on structuring our research as well as reviewing our paper.

References

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014a. Modelling sarcasm in twitter, a novel approach. In *WASSA@ACL*.

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014b. Modelling sarcasm in Twitter, a novel approach. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–58, Baltimore, Maryland. Association for Computational Linguistics.

Paula Carvalho, Luís Sarmento, Mário J. Silva, and Eugénio de Oliveira. 2009. 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*, TSA '09, page 53–56, New York, NY, USA. Association for Computing Machinery.

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

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.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

C. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A large self-annotated corpus for sarcasm.

Roger Kreuz and Gina Caucci. 2007. Lexical influences on the perception of sarcasm. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, pages 1–4, Rochester, New York. Association for Computational Linguistics.

Mohamed Lichouri, Mourad Abbas, Besma Benaziz, Aicha Zitouni, and Khaled Lounnas. 2021. Preprocessing solutions for detection of sarcasm and sentiment for Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 376–380, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Ya-Qin Liu, Cheng Wang, and Lu Zhang. 2009. Decision tree based predictive models for breast cancer survivability on imbalanced data. In *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, pages 1–4.

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.

Tomás Ptácek, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on czech and english twitter. pages 213–223.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.