Adding Humor and Sarcasm to Political Parody Detection - A Reproduction

Devin Ling
Rutgers University
dtl65@rutgers.edu

Ruan Jamolod
Rutgers University
rtj36@rutgers.edu

Abstract

This project is a reproduction of the multi-encoder model developed in *Combining Humor and Sarcasm for Improving Political Parody Detection* (Ao et al., 2022). We also build upon the original paper by exploring another way to combine the model. The model makes use of the linguistic features of humor and sarcasm to improve on the classification of parody for short text samples. We reproduced three parallel encoders for parody, humor, and sarcasm. The encoders were combined using different methods and the final model was investigated for improvement upon the vanilla binary classification model by Maronikolakis et al. (2020) among others. The source code can be found at our Github. [1]

1 Introduction

Today, political commentary is widely used in social media in order to push agendas and express opposition. Consequently, it is not uncommon to see parody expression within the political space (Davis et al., 2018). This type of expression may result in the spreading of false rumors and fake news. In order to combat this, NLP can be used to detect parody in social media. Current NLP applications require an improvement in parody detection, since social media (e.g., Twitter) has become increasingly more popular for politicians. Thus, there's a desire to better current methods for classification of parody. One route to take is to make use of common linguistic features that come with parody, specifically humor and sarcasm (Haiman, 1998; Highfield, 2016).

While parody detection may suffice as a binary classification task, it is important to consider linguistic features common to what we know as parody. This consideration of humor and sarcasm has now a higher level of parody analysis into this classification task. Since political parody detection is niche, it is important to expand upon via textual devices (e.g., figurative language).

Currently, in the NLP space, there is no way to combine humor and sarcasm to detect parody within social media. It is particularly challenging because of several reasons. There exists ambiguity and subjectivity in humor and sarcasm. They are both subtle and highly contextual literary devices, which can make it difficult for models to identify them (e.g., cultural references, current events, type of audience). There is also much variability in humor and sarcasm, since there is a wide range of jokes (e.g., irony, puns). Consequently, a model must be able to recognize and understand a wide range of linguistic patterns. Although this has been done in isolation previously, there has been no work done to jointly model the two *along with* identifying parody. There must first exist high-quality data sets for humor and sarcasm types. Then, a model must be able to utilize the large amount of information to its fullest.

As mentioned, each of these devices have been studied in isolation to parody. There has been work to model humor, which focuses on identifying jokes (Annamoradnejad and Zoghi, 2020; Taylor and MazIack, 2004) as well as work to model sarcasm, which focuses on identifying irony (Ghosh et al., 2021). There has also been a study on parody using vanilla transformer models that introduced a data set of political parody accounts on Twitter (Maronikolakis et al., 2020). However, overall, there has been little investigation into what parody actually *is* and how its linguistic features can be used to identify it. This area of work is what Ao et al. (2022) focuses on and extends upon to improve the current state of parody detection.

To combine humor and sarcasm in parody detection, Ao et al. (2022)'s approach is a trained

[1] https://github.com/devin-ling/Parody-Prediction-Multi-Encoder

and fine-tuned multi-encoder. The multi-encoder is used on a sample of tweets. It is made up of three parallel encoders – one for parody detection, one for humor classification, and one for sarcasm classification. This allows the first parody encoder to be enhanced with knowledge of what humor and sarcasm are. Three approaches are explored to combine the three encoders – concatenation, self-attention, and max pooling. We attempt at a fourth way – weighted combination. This helps with determining relevancy of each encoder. Results for the model are evaluated using F1 scores.

Summary of Contributions

- A reimplementation of a new model for political parody detection for Twitter (self-attention model), improving upon and outperforming the previous binary classification model by Maronikolakis et al. (2020).
- A look into the limitations of NLP applications for accurately capturing humor and sarcasm in parody via quantitative analysis.
- A look into the limitations of transfer learning for classification tasks.

2 Related Work

Due to the closeness of humor, sarcasm, and parody we can split related works into several fronts.

2.1 Parody

The first front being research on parody itself. Apart from the 2020 study focusing on political parody on Twitter (Maronikolakis et al., 2020) work on political parody found on social media is quite sparse. We can expand the scope of political parody to include work in the field of political satire. Work on this topic is largely related to the detection of misleading, or "fake," news in the media (Rashkin et al., 2017; Levi et al., 2019; McHardy et al., 2019). These studies vary in implementation with some opting to utilizer BERT models while others opting for LSTM models. Regardless, the success of these political satire detectors are rather high.

One clear distinction to be made is that parody and satire are not the same type of literature, as parody can be considered a subspecies of satire (Chatman, 2001). Satire encapsulates a larger genre not just limited to texts (a topic that can be further studied in NLP). Another distinction that should be made is that the related studies mentioned do not specifically relate to social media, a platform that is still being significantly researched.

2.2 Humor

Humor is a topic that is studied more within NLP, with various humor recognition studies in academia. These studies range from textual humor to even visual humor (Weller and Seppi, 2019; Hasan et al., 2019; Chen and Soo, 2018). However, humor on social media is not studied as widely. One such study focuses on Facebook-post reactions as it relates to humor (Yang et al., 2021). Another study utilizes low-resource languages (specifically Telugu) to detect humor (Bellamkonda et al., 2022).

2.3 Sarcasm

Sarcasm, as compared to the previous two topics, is fairly well studied in NLP with a good bit of research being done regarding social media (Oprea and Magdy, 2020; Ghosh and Veale, 2016; Cai et al., 2019). These studies seem to follow the same models as previously discussed, using CNNs and LSTMs to properly classify sarcasm

3 Method

The original multi-encoder model was reproduced from scratch beginning with the three individual text encoders. The encoders were combined and the final representation was passed through a classification layer. From there, we replicated the quantitative analysis using F1 scores.

3.1 Data

For the main parody data set, we use the same corpus as in the original paper and introduced by Maronikolakis et al. (2020). It contains 131,666 tweets, where 65,956 tweets are from parody political accounts and 65,710 tweets are from real political accounts. We utilize a 80-10-10 train-validation-test split in order get the most out of our training.

We use only two of the three data sets provided due to time constraints. First, the person data set, or all tweets randomly split into train, dev, test was used. Second, the gender split was used – two different splits based on gender (male / female) (e.g., male for train/dev and female for test, female for train/dev and male for test).

3.1.1 The Individual Encoders

There are three main data sets used in training the individual encoders and the entire model – one for parody, one for humor, and one for sarcasm. All encoders use BERTweet (Nguyen et al., 2020), a BERT based pre-trained on English tweets.

Parody Encoder The parody encoder takes the regular pre-trained BERTweet model (Nguyen et al., 2020) and finetunes it on the parody data set discussed in 3.1.

Humor Encoder We use the data set introduced by Annamoradnejad and Zoghi (2020), which contains a large corpus of English short texts classified as humorous or non-humorous. Then, we pre-trained the BERTweet model on 10,000 random humor-only texts using masked language modeling. The final encoder is finetuned on 40,000 random texts as a humor classification task.

Sarcasm Encoder We use the English tweet data set introduced by Abu Farha et al. (2022). The data set consists of 1,267 sarcastic and 4,868 non-sarcastic tweets, also split 80-10-10 between training, validation, and testing. The original paper splits up the data into separate tasks, but we reformat the data to fit our uses. We pre-trained the BERTweet model on all sarcasm-only texts using masked language modeling. The final encoder is finetuned on all sarcasm tweets and non-sarcasm tweets.

Note that Ao et al. (2022) utilizes two sarcasm annotated datasets from Oprea and Magdy (2020) and Rajadesingan et al. (2015). However, the publicly available datasets lack the actual tweets themselves.[2]

3.2 Fusion Layer

By extracting the 'classification' [CLS] token from each encoder respectively we obtain representations for each encoder. For each encoder output, $f \in \mathbf{R}^{768}$. We then combine the outputs of each encoder in the following four ways.

Concatenation The three text representations are simply concatenated together.

Self-attention 4-head self-attention is applied for each output to find correlations between each representation.

Max-pooling Max-pooling is applied to each dimension of each output for a result of $f \in \mathbf{R}^{768}$

Weighted Combination We applied weights to each encoder and took the weighted sum of each output. This method serves to determine the relevancy of each encoder and support the results from self-attention.[3]

3.3 Classification Layer

Finally, the full combined representation is passed through a classification layer via a sigmoid function for determining whether a post is parody or not.

3.4 Implementation Details

Note that different hyperparameters were tried for when training the encoders with optimality.

Humor Encoder For pre-training, the batch-size is set to 16 and the number of training epochs is set to 2 with a learning rate of $2e^{-5}$. For humor classification, we use batch size of 32 and the number of epochs is set to 2 with a learning rate of $3e^{-5}$.

Sarcasm Encoder For pre-training and classification, the same hyperparameters were used as for the humor encoder.

Multi-encoder For the combined multi-encoder, we use a batch size of 32 and the learning rate is set to $2e^{-5}$. The entire model is fine-tuned for 2 epochs. Note that these hyperparameters are just for the best performing model.

3.5 Evaluation

All models tested are evaluated using F1 scores. They are each run once due to time constraints.

4 Experiments

The results of our tested models on the two splits are shown in Tables 1 and 2. Our results have smaller F1 values as compared to the paper's results due to our encoders being trained with smaller amounts of data. However, what we find is that our results are consistent despite the small difference.

[2]After contacting the original authors of the papers we were able to obtain the new sarcasm annotated dataset based off Oprea and Magdy (2020), although the data set was smaller in size.

[3]Originaly, we wanted to further integrate knowledge from political texts other than social media tweets (e.g., political speeches, campaign slogans), but we could not get access to a labeled data set in time.

Person	
Model	**F1 Score**
Single Encoder	
BERT	86.12
RoBERTa	88.91
BERTweet	89.98
Multi-Encoder	
Concatenation	88.22
Self-attention	**91.01**
Max-pooling	90.88
Weighted Combination	90.98

Table 1: F1 Scores of each model for Person split

Gender		
Model	**M →F**	**F →M**
Single Encoder		
BERT	84.81	83.21
RoBERTa	86.12	83.11
BERTweet	88.00	85.01
Multi-Encoder		
Concatenation	85.12	81.84
Self-attention	**88.41**	**87.53**
Max-pooling	86.09	84.98
Weighted Combination	87.09	85.28

Table 2: F1 Scores of each model for the Gender split

For the single-encoder models, BERTweet does the best, which is expected. For the multi-encoder models, self-attention does the best. Generally, the multi-encoder model outperforms the single encoder models, which indeed suggest that combining parody detection with humor and sarcasm is beneficial. Interestingly, we see that weighted combination does well for itself, beating out max pooling and concatenation by a small amount.[4] Concatenation actually reduces the quality of performance, since it weighs encoders equally.

As for the gender split[5], we see that, when training on female accounts, the model scores a higher F1 score by about 1. This is most likely because there are less female politicians, and thus, less tweets to train and possibly overfit on.

As in the paper, we performed an ablation study on different combinations of the encoders – parody with humor (P+H) and parody with sarcasm (P+S). We specifically tested the self-attention model on the person split in the interest of time. The results can be seen below in Table 3.

Again we find that our results have smaller F1 values, but still maintaining the original findings

[4]We saw that the sarcasm encoder had the most relevance out of the three.

[5]M →F model was trained on female Twitter accounts and F →M was trained on male Twitter accounts

Person	
Model	**F1 Score**
Single Encoder	
BERTweet	89.98
Multi-Encoder	
P + H	90.25
P + S	90.55
P + H + S	**91.01**

Table 3: F1 Scores for Different Combinations of the Self-attention Model on the Person split

of the paper. We see that P+S combination does slightly better overall. This most likely has to do with corpus being political parody, as political parody has more sarcastic features. Nonetheless, the model has the most gains when combining both sarcasm and humor.

5 Conclusions

We aimed to accomplish a few things in this project: retrain our own versions of sarcasm and humor encoders, and to test our own way of combining the encoders. We train each encoder-model on the parody data set and then pass it through a classification layer.

This approach provides an efficient way to detect parody on social media. Due to the lack of availability of publicly labeled tweets, and the limitations in tweet acquisition, this approach could possibly be even more accurate with the collection of more data. On top of this, due to our lack of computational resources we were not able to test as much as we wanted to.[6] Regardless, we gained insight into transfer learning with specific linguistic features and were able to see how successful a model like this can be when trying to classify parody in social media.

References

Ibrahim Abu Farha, Silviu Vlad 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)*, pages 802–814, Seattle, United States. Association for Computational Linguistics.

Issa Annamoradnejad and Gohar Zoghi. 2020. Colbert: Using bert sentence embedding for humor detection. *arXiv preprint arXiv:2004.12765*.

[6]It had also taken a while for us to implement the encoders, as there was no reference code to help us.

Xiao Ao, Danae Sanchez Villegas, Daniel Preotiuc-Pietro, and Nikolaos Aletras. 2022. Combining humor and sarcasm for improving political parody detection. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Sriphani Bellamkonda, Maithili Lohakare, and Shaswat Patel. 2022. A dataset for detecting humor in Telugu social media text. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 9–14, Dublin, Ireland. Association for Computational Linguistics.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multi-modal sarcasm detection in Twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515, Florence, Italy. Association for Computational Linguistics.

Seymour Chatman. 2001. Parody and style. *Poetics Today*, 22(1):25–39.

Peng-Yu Chen and Von-Wun Soo. 2018. Humor recognition using deep learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 113–117, New Orleans, Louisiana. Association for Computational Linguistics.

Jenny L Davis, Tony P Love, and Gemma Killen. 2018. Seriously funny: The political work of humor on social media. *New Media Society*, 20(10):3898–3916.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, San Diego, California. Association for Computational Linguistics.

Debanjan Ghosh, Ritvik Shrivastava, and Smaranda Muresan. 2021. "laughing at you or with you": The role of sarcasm in shaping the disagreement space. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, page 1998–2010.

John Haiman. 1998. *Talk is cheap: Sarcasm, alienation and the evolution of language*. Oxford University Press.

Md Kamrul Hasan, Wasifur Rahman, AmirAli Bagher Zadeh, Jianyuan Zhong, Md Iftekhar Tanveer, Louis-Philippe Morency, and Mohammed (Ehsan) Hoque. 2019. UR-FUNNY: A multimodal language dataset for understanding humor. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2046–2056, Hong Kong, China. Association for Computational Linguistics.

Tim Highfield. 2016. News via voldemort: Parody accounts in topical discussions on twitter. *New Media amp; Society*, 18(9):2028–2045.

Or Levi, Pedram Hosseini, Mona Diab, and David Broniatowski. 2019. Identifying nuances in fake news vs. satire: Using semantic and linguistic cues. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 31–35, Hong Kong, China. Association for Computational Linguistics.

Antonios Maronikolakis, Danae Sánchez Villegas, Daniel Preotiuc-Pietro, and Nikolaos Aletras. 2020. Analyzing political parody in social media. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 4373–4384.

Robert McHardy, Heike Adel, and Roman Klinger. 2019. Adversarial training for satire detection: Controlling for confounding variables. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 660–665, Minneapolis, Minnesota. Association for Computational Linguistics.

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. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 1279–1289.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark. Association for Computational Linguistics.

J.M. Taylor and L.J. MazIack. 2004. Humorous wordplay recognition. *IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*.

Orion Weller and Kevin Seppi. 2019. Humor detection: A transformer gets the last laugh. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3621–3625, Hong Kong, China. Association for Computational Linguistics.

Zixiaofan Yang, Shayan Hooshmand, and Julia Hirschberg. 2021. CHoRaL: Collecting humor reaction labels from millions of social media users. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4429–4435, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.