Making Your Tweets More Ironic

Toma Puljak, Rene Kustura, Kristian Djaković

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{author}@fer.hr

Abstract

Irony, by itself, is complicated and detecting irony in tweets is a difficult task. In this paper, we analyse tokens extracted from ironic tweets and find which are considered most ironic. To choose which tokens to analyse, we looked at attention values produced by the RoBERTa model when classifying tweets. The selected tokens were then appended to selected tweets from train and test sets from the SemEval2018 Task3 dataset. From the results, by comparing the negative change in the non ironic sentiment, we conclude that the most influential token and token pairs are ”great” and ”Success ␣” respectively.

1. Introduction

Irony is complicated. In general, irony refers to a clash between the expectation and outcome of an expression. (Kreuz, 2020). While this definition looks simple enough, detecting irony in text proves to be a difficult task, not only for models but for people as well. The goal of this paper is to make tweets more ironic by adding ”ironic” tokens to non ironic tweets. We find ironic tokens by analysing attention values outputted from a classification model.

The attention mechanism allows modeling of dependencies without regard to their distance in the input or output sequence (Vaswani et al., 2017). What's great about attention is that it adds a layer of explainability to the model. Attention values can be used to interpret the focus of a model when completing various tasks. In this paper, attention values are used to gather the most ironic tokens from data samples. The most ironic tokens are those with the highest attention values.

For classification purposes, we used a pretrained RoBERTa Transformer model that was trained on the SemEval2018 Irony Detection dataset (Liu et al., 2019). The model uses multi-head attention, thus we used the *attention rollout* method to aggregate attention values across heads for single tokens and connections between token pairs.

Once we gathered the most ironic tokens, we applied them to tweets sample from the training and testing datasets and observed the change between the non-ironic prediction confidence score. We use this change in prediction confidence score as a metric to evaluate how much a token, or token pair, impacts the ironic sentiment of a given tweet.

2. Background

In this section, we provide an overview of the classification model setup, attention aggregation methods, and the datasets to evaluate our analysis goal.

2.1. Model Setup

For classification purposes, we used a pretrained RoBERTa Transformer model that had the best performance in irony detection. The model can be found here: `https://huggingface.co/cardiffnlp/twitter-roberta-base-irony`. RoBERTa is a retrained BERT model with improved performance. The details of the architecture can be found in (Liu et al., 2019).

2.2. Attention as an Interpretation Metric

Using attention as an interpretation metric is arguable and research by (Serrano and Smith, 2019) shows that it doesn't necessarily produce correct importance ranking. Nonetheless, we use attention values to interpret how ironic are the tokens, or token pairs.

We considered the alternative of analysing all tokens from selected ironic tweets, but that would require a lot of computation time which we leave for future work.

2.3. Attention Aggregation

To aggregate attention values into a singular output, we use the *attention rollout* method. Attention rollout is a recursive method that calculates the product of all attention weights in the attention graph. (Abnar and Zuidema, 2020) This method will compute a singular output for a single token that we feed into it and if we squeeze the resulting matrix without extraction, we get attention values for two connected tokens in the input sentence.

2.4. Data

The dataset used to train the RoBERTa model for irony detection is the SemEval2018 Irony Detection dataset (Hee et al., 2018) which is split into training and testing sets. Each tweet is annotated with a binary label of 0 or 1, representing a non-ironic and ironic label respectively.

We applied a simple preprocessing step for all tweets which changes all user tags to *@user* and removes all URLs. This step ensures that tagging specific users in tweets doesn't affect the tokenization and that URLs don't affect the model at all because they, by themselves, do not add any semantic value to the tweet.

3. Analysing the Most Ironic Tokens

In this section, we will cover the process of finding the most ironic tokens from the used dataset and the methods used for evaluating the results.

3.1. Finding the Tokens

The first step in finding the tokens is to extract all tweets that were annotated as ironic from the training and testing datasets, keeping them separate so we can test the results of the experiment separately. We then classified those tweets using the RoBERTa model to obtain a confidence score for each label.

By sorting the tweets, by their ironic label confidence score, we obtained a list of the most ironic tweets in the datasets, according to the model. The five most ironic tweets from the training and testing datasets can be seen in Table 1 and 2 respectively.

Once we obtained the list of the most ironic tweets, we performed attention aggregation methods outlined in Section 2.3. and sort the tokens by the attention score in descending order. That gives us a list of singular tokens and token pairs that the model focused on more while performing classification. A subset of obtained singular and token pairs for the tweet ”Yay for getting pink eye again! #whyme” can be seen in Table 3 and 4 respectively.

3.2. Adding Ironic Tokens to Tweets

For tweets which we will analyse, we randomly sampled 100 tweets, from the train and test datasets.

To add the ironic tokens into the text we simply appended them to the end of the tweet. This was done for each token in the list applied to each tweet in the subset.

We are aware that simply appending the token, or tokens, to the end of a tweet may generate something meaningless but ensuring that the newly formed tweet is semantically correct is out of the scope of this paper and we leave that for future work.

4. Results

4.1. Single Token

We analysed 100 tweets paired with the top 10 tokens from the top 10 most ironic tweets. This produced 9200 and 9600 new tweets from the train and test sets respectively. Because some tweets had less than 10 tokens, there were only 92 and 96 tokens to append in total.

A sample of the top 3 and bottom 3 changes in non-ironic confidence scores in the train can be seen in Table 5, with the full results available at https://github.com/Tpuljak/tar_irony_detection/blob/main/results/added_irony_train_singles.txt?raw=true.

The results show that the token with the most influence on the ironic sentiment was ”great” and that token ”where” emphasizes the non ironic sentiment of a tweet.

The top three changes in ironic sentiment belong to the token ”great” which is a satisfying result because we, as people, can interpret ”great” ironic if used in the right context.

The bottom three changes show that tweet ”Why I love penguins of Madagascar #humor #reliefcomedy” is easily influenced by adding an ironic token to the end. Even though the appended tokens to not represent any semantic meaning, we managed to flip the sentiment of the RoBERTa classifier.

For brevity, we will not analyse test set results here, but only mention that the top and bottom token pairs were, again, ”great” and ”how” respectively. The full test set results can be seen here https://github.com/Tpuljak/tar_irony_detection/blob/main/results/added_irony_test_singles.txt?raw=true.

4.2. Token Pairs

We analysed 100 tweets paired with the top 10 pairs of tokens from the top 20 most ironic tweets, producing a total of 20 000 new tweets. A sample of the top and bottom 3 changes in non-ironic confidence scores in the train set can be seen in Table 6, with the full results available at https://github.com/Tpuljak/tar_irony_detection/blob/main/results/added_irony_train_pairs.txt?raw=true.

The results show that the token pair with the most influence on the ironic sentiment was ”Success ␣” and that token pair ”in It” emphasizes the non ironic sentiment of a tweet.

Taking a look and the entire result list, we see that the tweet ”Why I love penguins of Madagascar #humor #reliefcomedy” has been greatly influenced by the appended tokens. Even though the tweet was actually labeled as ironic before appending the tokens, adding ironic tokens flipped its sentiment to non ironic. This is in line with the results from the previous section.

Regarding the tokens that most influenced the ironic sentiment, it is interesting to see token pair ”wow ␣”. Intuetively, ”wow” can be seen as a sarcastic or ironic token from a semantic perspective, thus it is expect that it would change the ironic sentiment of a tweet.

For brevity, we will not analyse test set results here, but only mention that the top and bottom token pairs were ”Thanks @” and ”up be” respectively. The full test set results can be seen here https://github.com/Tpuljak/tar_irony_detection/blob/main/results/added_irony_test_pairs.txt?raw=true.

5. Future Work

Even though we have managed to change ironic sentiment by applying ironic tokens to tweets, we propose a more robust approach that takes into account that the semantics of a tweet remains intact after adding the token. Additionally, possible exploration of adding tokens at arbitrary points in the tweet, not only at the end.

As mentioned, a higher attention score doesn't necessarily mean that a token is more ironic, thus in future work, we could explore different metrics for obtaining tokens that we want to analyse.

6. Conclusion

Results have shown that the most influential token and token pairs are ”great” and ”Success ␣”. By simply appending them to the end of a tweet we managed to change the ironic sentiment as judged by the RoBERTa model.

The results for the single token experiment are very satisfying because the token ”great” does have ironic meaning if used in the right context.

Table 1: Most Ironic Tweets in the Training Set

Tweet	Confidence score
Yay for getting pink eye again! #whyme	1
@user yay!!!! It works #HateWhenThingsDontWorkRight	0.99
@user great Christmas.	0.99
Well done @user for making it possible to get emergency messages to a member of staff.	0.99
Isn’t it great to sleep 5 hours and feel like a million bucks? #gettingold	0.99

Table 2: Most Ironic Tweets in the Test Set

Tweet	Confidence score
Just great when you’re mobile bill arrives by text	1
OH and now the District line has major signal failures and delays FANTASTIC!!!	0.99
A wonderful day of starting work at 6am	0.99
Having to be up in four hours sounds great	0.99
@user @user @user @user @user nice to see the ambulance service is so important to OUR mps	0.99

Table 3: Single Tokens With Highest Attention for the Tweet ”Yay for getting pink eye again! #whyme”

Token	Attention score
me	1
Y	0.37
pink	0.33
ay	0.29
why	0.14

Table 4: Token Pairs With Highest Attention for the Tweet ”Yay for getting pink eye again! #whyme”

Token1	Token2	Attention score
ay	Y	0.6
#	Y	0.6
␣	Y	0.6
me	Y	0.6
again	Y	0.6

Table 5: Sample of Single Token Results on the Train Set

Index	Tweet	Added Token	New Non-Ironic Score	Change
1	@KyleCGShore a kiss on the cheek next time is enough :)	great	0.0644627	0.871908
2	@sophiaqualquer what for ? waste of time...:)	great	0.099291	0.833875
3	Is holding not a rule anymore?	great	0.091083	0.825079
9198	Why I love penguins of Madagascar #humor #reliefcomedy	for	0.685702	-0.522685
9199	Why I love penguins of Madagascar #humor #reliefcomedy	in	0.702336	-0.539319
9200	Why I love penguins of Madagascar #humor #reliefcomedy	where	0.722789	-0.559772

Table 6: Sample of Token Pair Results on the Train Set

Index	Tweet	Added Tokens	New Non-Ironic Score	Change
1	Gah! Desperately trying to bust this cold!	Success ␣	0.0267051	0.939541
2	Gah! Desperately trying to bust this cold!	what great	0.0322938	0.933953
3	@BlogOfErised so I gave up and am now heading for bed orz	wow ␣	0.0543532	0.918367
19797	Why I love penguins of Madagascar #humor #reliefcomedy	music It	0.715983	-0.552966
19798	Why I love penguins of Madagascar #humor #reliefcomedy	already @	0.722912	-0.559895
19799	Why I love penguins of Madagascar #humor #reliefcomedy	in It	0.733982	-0.570964

Even though the most influential token pair seems to be meaningless when appended to tweets, it is interesting to see tokens like ”wow” and ”great wow” have a great influence on the ironic sentiment.

References

S. Abnar and W. Zuidema. 2020. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*.

C. Van Hee, E. Lefever, and V. Hoste. 2018. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana, June. Association for Computational Linguistics.

Roger J. Kreuz. 2020. What makes something ironic? *The Conversation*, February.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

S. Serrano and N. A. Smith. 2019. Is attention interpretable? *arXiv preprint arXiv:1906.03731*.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.