Differentiating Irony and Sarcasm: A Challenge in NLP

Bojan Puvača, Florijan Sandalj, Ivan Unković

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{bojan.puvaca, florijan.sandalj, ivan.unkovic}@fer.hr

Abstract

Forms of dishonest speech like irony and sarcasm present significant challenges in natural language processing (NLP) due to their inherently complex and context-dependent nature. Moreover, the thin line between irony and sarcasm is often defined with inconsistency, which brings into question its usefulness. In this paper, we take a look at how the difference between irony and sarcasm is defined in the field of NLP, how different models perform on different tasks of their detection, how and where those tasks overlap and finally, try to conclude does solving those tasks separately even make practical sense.

1. Introduction

Irony and sarcasm being the main representatives of dishonest speech makes their detection a mainstay problem in Natural Language Processing (NLP). However, their relationship often confuses researchers, as the way they are defined in various datasets and research papers is far from consistent. Analyzing the results of our research, we hope to shed some light on this issue by answering some key questions regarding the way problems based on irony and sarcasm should be tackled in further research.

Our goal is to determine the practical usefulness of the currently most popular definition of irony and sarcasm in NLP, which states that sarcasm is a sharp, mean-spirited form of irony. We conduct a meta-analysis of the performance of different models on the tasks of irony and sarcasm detection in order to determine the amount of overlap between the two tasks and whether or not this definition makes sense in practice by examining how well models trained on one task perform on the other. Our *code*[1] is publicly available.

As both irony and sarcasm can be forms of dishonest speech, we hypothesize that models that perform well on one task will also perform well on the other, in which case the current distinction between the two wouldn't be of much use, and we would be better off either treating the detection of the two as the same task or creating a more useful distinction between them in the context of NLP.

In the discussion section, we aim to answer the following questions that could aid future research of irony and sarcasm in NLP:

- How much overlap is there between irony and sarcasm in the most popular datasets?
- Does separating irony and sarcasm detection make sense in practice?
- Are there models better suited for one task than the other?
- Should the terms "irony" and "sarcasm" be redefined in the context of NLP to make them more useful for research?

[1] https://github.com/bpuvaca/irony-detection-tar2024

2. Irony and sarcasm in NLP

This chapter takes a look at the related relevant works in NLP that deal with irony and sarcasm, and introduces our own view of the problem.

2.1. Related work

The datasets we found to be the most popular and relevant in irony and sarcasm detection research are iSarcasm (Oprea and Magdy, 2020) and SemEval (Van Hee et al., 2018). In iSarcasm, the collected tweets were classed into five categories of ironic speech, two of which were useful for this research:

1. *Sarcasm*: tweets that contradict the state of affairs and are critical towards an addressee
2. *Irony*: tweets that contradict the state of affairs but are not obviously critical towards an addressee

In SemEval-2018, two datasets are presented: taskA, where any form of ironic speech labeled 1, while regular speech is labeled 0, and taskB, where the ironic speech is further classed into three categories:

1. *Verbal irony by polarity*: instances containing an expression whose polarity (positive, negative) is inverted between the literal and the intended meaning
2. *Situational irony*: instances describing situational irony, or situations that fail to meet some expectations
3. *Other verbal irony*: instances that show no polarity contrast between the literal and the intended meaning, but are nevertheless ironic

In both datasets the tweets were manually annotated after being collected using specific hashtags, such as #irony, #not and #sarcasm.

From actual attempts at classifying ironic speech, it's worth mentioning (Nguyen et al., 2020), where the proposed BERTweet architecture uses a modified pre-training procedure based on RoBERTa (Liu et al., 2019) and is pre-trained on a corpus of tweets, making it more suitable for tweet classification tasks. It was tested on the SemEval-2018 taskA and produced competitive results. Also, in

(Potamias et al., 2020), the authors build on the RoBERTa model to tackle irony classification, which proved to be a success. In (Tomás et al., 2023), images associated with the tweets are also used for classification. However, none of these works explored the differences between irony and sarcasm and whether it is useful to treat them as separate concepts, which is the direction this paper takes.

2.2. Sarcasm - irony's meaner cousin

The online Merriam-Webster dictionary defines sarcasm as "a sharp and often satirical or ironic utterance designed to cut or give pain" (Merriam-Webster, 2024). The iSarcasm dataset (Oprea and Magdy, 2020) aligns with this definition, as the "sarcasm" label is a subset of the unfortunately named "sarcastic" label, which actually indicates any kind of ironic speech.

In this context, irony refers to any type of speech that is based on polarity - whether that be pointing out the polarity between the expected and actual outcome of a situation (situational irony, in which case there is no dishonest speech), or expressing with words the opposite of what we mean (verbal irony based on polarity). On the other hand, sarcasm refers to instances that show no polarity contrast between the literal and the intended meaning, but are nevertheless ironic (Van Hee et al., 2018). Although this definition works on paper, there are some pitfalls. Most notably, the line between sarcasm and verbal irony is unclear, as whether or not a statement is mean-spirited is subjective. Statements with a mean tone, but without polarity can also be considered sarcastic, meaning that sarcasm doesn't necessarily entail irony. Also, tweets that contain irony and aren't directed at a specific person can still be considered sarcastic, as they often target a group of people, concepts, ideas or themselves in the form of self-deprecating humor. How the object of the irony affects the classification is another unanswered question.

All things considered, the iSarcasm (Oprea and Magdy, 2020) dataset does a solid job at distinguishing between the two. However, the usefulness of this distinction is somewhat questionable, as both concepts can be based on dishonest speech, meaning that in practice there might not be much use in distinguishing between them.

In this paper, we will take a closer look at how different models perform on the seperate tasks of irony and sarcasm detection, with the goal of determining the amount of overlap between the two tasks and the potential benefits of redefining their distinction. We will do so using a combination of the iSarcasm (Oprea and Magdy, 2020) and the SemEval-2018 (Van Hee et al., 2018) datasets, both of which contain tweets that are labeled as either ironic or sarcastic based on this distinction.

3. Experimental setup

Three seperate datasets were created for the experiment of comparing irony and sarcasm detection, one containing tweets labeled as ironic, one containing tweets labeled as sarcastic and the third one combining the first two.

All three datasets were constructed as a binary classification task, with neutral tweets, not containing any irony or sarasm, being labeled as negative, and the tweets containing either irony or sarcasm being labeled as positive.

As the SemEval-2018 dataset contained significantly more ironic tweets than sarcastic ones, we merged it with the iSarcasm dataset in order to produce larger and more balanced datasets. We found this approach to be justified, as both datasets discerned between irony and sarcasm in a similar manner, both in their explanations for the labels and upon manual inspection of the tweets.

Various models were trained on all three of these tasks, after which their performance was evaluated on the test sets of all three datasets in order to determine the amount of overlap between irony and sarcasm detection.

The same models were also trained and evaluated on the unmodified SemEval-2018 dataset, in order to showcase their performance on a standard dataset for ironic speech detection.

In section 3.1., we will describe the process of constructing the datasets and in 3.3. we will describe the models used in this experiment.

3.1. Construction of the datasets

The positively labeled tweets for the sarcasm detection task were taken from both the iSarcasm and SemEval-2018 datasets, using tweets labeled as "sarcasm" and "other verbal irony", respectively. Although their definitions seem different, when manually inspecting the contents of those tweets, we found that they almost exclusively contained the hashtag #sarcasm and contained sarcastic remarks.

For the irony detection task we used the same method, only this time using the tweets labeled as "irony" from the iSarcasm dataset, and the tweets from the first and second category from the SemEval-2018 dataset, meaning "verbal irony by means of a polarity contrast" and "situational irony" respectively. All three of these categories were based on a polarity contrast, which is why they were grouped together.

Negative examples for both tasks were taken from both the iSarcasm and the SemEval-2018 datasets, using tweets labeled as not being sarcastic or ironic.

The reason for not including sarcastic tweets as negative examples in the irony detection task and vice versa is so we can effectively use models trained on one task for some other task, as otherwise the models would be trained to label the positive examples of the other task as negative.

The combined dataset was created by merging the sarcastic and ironic datasets, with the positive examples of both tasks being labeled as positive, and the neutral examples being labeled as negative.

All three datasets were split into training, validation and test sets using a 60/20/20 split and undersampled in runtime to ensure that all three tasks had the same number of positive and negative examples in all three sets in order to ensure their fair comparison. For the combined dataset, we also considered its unbalanced variant with all the available data. This was done to see if the approach of seperating sarcastic and ironic data makes sense in the general case, or if there is enough overlap between the two concepts to treat irony and sarcasm as one task if we aren't concerned with the distinction between the two.

3.2. Tweet preprocessing

Because of the specific language used in tweets, as well as the presence of hashtags, links and mentions, we used the tweet normalization method proposed by the authors of BERTweet (Nguyen et al., 2020) in order to preprocess the tweets. This method performs subword tokenization and replaces mentions and links with special tokens. Although this method was proposed to be used with the BERTweet model, we found it's use to be beneficial for all models used in this experiment.

The hashtags used to find and scrape the tweets in the SemEval-2018 dataset were removed from the tweets, as not to make the detection task trivial.

3.3. Models

As our goal was to compare the performance of state-of-the-art models on the tasks of irony and sarcasm detection, we focused on transformer-based models. Specifically we used the BERT (Devlin et al., 2019) and BERTweet (Nguyen et al., 2020) models, as they showed excellent performance on the SemEval-2018 task in previous research (Potamias et al., 2020; Nguyen et al., 2020).

Both of these models were used with their out-of-the-box sequence classification configurations, fine-tuned on our different tasks (Wolf et al., 2020). We will refer to these models as BERT and BERTWEET.

We have also used BERT and BERTweet as encoders and fed their outputs to a bidirecitonal LSTM layer from the PyTorch library (Paszke et al., 2017), followed by a dropout layer and a dense layer which performs the classification. We will refer to these models as BERT+LSTM and BERTWEET+LSTM.

A convolution based approach was used in a similar manner, with two convolutional layers replacing the LSTM layer. In these models, the convolutional layer was followed by a max pooling layer, a dropout layer and a dense layer. We will refer to these models as BERT+CNN and BERTWEET+CNN.

All of these models were trained on the three tasks described in section 3.1. and the SemEval-2018 task, along with a simple baseline model based on an LSTM network and GloVe embeddings, which we will refer to as the BASELINE model.

3.4. Training setup

All of the transformer-based models were trained using cross-entropy loss, the AdamW PyTorch optimizer and a learning rate scheduler with a warmup. The hyperparameters were optimized on a per-model basis using a grid search over the combined and balanced dataset. The baseline model was trained using only the Adam optimizer and cross-entropy loss.

Models were trained for up to 10 epochs, with batch size of 16 and early stopping based on the F1 score on the validation set. For each task and model pair, 5 different runs were performed, with the results being averaged in the end.

3.5. Evaluation

All of the afformentioned models trained on the SemEval-2018 training set were evaluated on the appropriate SemEval-2018 test set, while the models trained on each of our three tasks were evaluated on all three of our test sets. The F1 metric was used as the primary metric for evaluating and comparing the different models.

4. Results

In this section, we will show the results on all of the models on the SemEval-2018 dataset (section 4.1.). After that, an analysis of the performance of the best models on our three tasks and the transferability of the models between the tasks will be presented in section 4.2.. Performance of all models on irony and sarcasm detection tasks will be analyzed in section 4.3. in order to determine the correlation between the model performances on the two tasks.

4.1. SemEval-2018 dataset

The results of the models on the SemEval-2018 dataset can be seen in Table 1. The models based on BERTweet outperformed the models based on BERT, which is in line with the results of previous research (Nguyen et al., 2020).

Our models with additional LSTM and CNN layers came close to the performance of the BERTWEET model and outperformed the baseline and BERT-based models. Based on these results, the models we will consider for the transferabilty tests are the BERTWEET and the BERTWEET+LSTM models.

Table 1: Results of the models on the SemEval-2018 dataset

Irony classification model	F1	Acc
BASELINE	0.625	0.645
BERT	0.654	0.654
BERTWEET	**0.785**	**0.788**
BERT+LSTM	0.657	0.657
BERTWEET+LSTM	0.763	0.768
BERT+CNN	0.669	0.670
BERTWEET+CNN	0.752	0.754

4.2. Irony and sarcasm transferabilty results

Tables 2 and 3 show the results of the BERTWEET and BERTWEET+LSTM trained on the three tasks, as well as the unbalanced version of the combined task. For each model and training setup, we showcase the F1 score on each of the three tasks.

	Irony	**Sarcasm**	**Mixed**
Irony	0.720	0.634	0.657
Sarcasm	0.622	0.710	0.682
Mixed	0.681	0.767	0.723
Mixed unbalanced	0.659	0.772	0.726

Table 2: F1 score matrix for model BERTWEET on the three tasks. Rows represent the task the model was trained on, while columns represent the task the model was evaluated on

	Irony	Sarcasm	Mixed
Irony	0.759	0.682	0.720
Sarcasm	0.547	0.743	0.660
Mixed	0.698	0.775	0.736
Mixed unbalanced	0.666	0.745	0.719

Table 3: F1 score matrix for model BERTWEET+LSTM on the three tasks

4.3. Irony and sarcasm performance analysis

The Table 4 shows the F1 scores of all the models on the tasks of irony and sarcasm detection. For this analysis the models were simply trained and evaluated on the tasks of irony and sarcasm detection. This data will be used to determine the correlation between model performance on the two tasks.

Table 4: Irony and sarcasm detection F1 scores

Model	Irony F1	Sarcasm F1
BASELINE	0.631	0.548
BERT	0.713	0.703
BERTWEET	0.720	0.710
BERT+LSTM	0.704	0.668
BERTWEET+LSTM	0.759	0.743
BERT+CNN	0.697	0.706
BERTWEET+CNN	0.767	0.746

5. Discussion

For the discussion part of this paper, we will answer the questions posed in the introduction, based on the results.

5.1. Overlap between irony and sarcasm detection

The results shown in tables 2 and 3 show solid transferability between the tasks of irony and sarcasm detection, with the BERTweet based models trained on one task performing decently well on the other.

The dropoff in performance when training on one task and evaluating on the other goes both ways, with the dropoff in both cases being of a similar magnitude. This suggests that framing sarcasm as a subset of irony in NLP doesn't tell the whole story, as in that case we would expect larger dropoffs in performance when training on the sarcasm task and evaluating on the irony task than the other way around.

This can be explained by the fact that while clearly not all ironic statements are sarcastic, a lot of the statements we perceive as sarcastic due to their tone don't actually contain any polarity and are therefore not picked up by the models trained on the irony detection tasks as ironic. In that case, irony and detection would have more of an overlapping relationship than a hierarchical one.

5.2. Usefulness of irony and sarcasm data separation

The results of the models trained on the combined dataset show excellent performance on the combined task, with the unbalanced variant with more data not showing much of a performance increase. Surprisingly, the models trained on the combined dataset dominated the task of sarcasm detection, even more so than the models trained on the sarcasm detection task, while irony detection proved to be more challenging.

Nevertheless, the results of the models trained on the combined dataset show that if our goal is to simply detect dishonest speech in general, we shouldn't be bothered with seperating irony and sarcasm in our datasets, as this approach doesn't seem to be beneficial when compared with combining the two tasks.

5.3. Suitability of different models for irony and sarcasm detection

The F1 scores obtained for models trained and tested on both irony and sarcasm indicate minimal differences between the tasks. To further test the similarity between the tasks, we calculated the correlation between the F1 scores of our models using the pearson coeffiecient.

$$r = 0.9421$$

The coeffiecient indicates that models performing well in irony detection also demonstrate strong performance in sarcasm detection and vice versa. Consequently, there appears to be no justification for the utilization of separate models for detecting irony and sarcasm, as the tasks exhibit strong similarity.

5.4. Redefinition of "irony" and "sarcasm" in NLP

Developing more distinct definitions of irony and sarcasm could be more beneficial in future research in NLP due to the shown similarities in their respective classification tasks. For example, the term "irony" could be defined exclusively as situational irony, whereas the term "sarcasm" could refer to all kinds of speech characterized by polarity. Alternatively, even less emphasis could be placed on their distinction, treating irony and sarcasm detection as closely related sequence classification tasks.

We would like to see these suggestions considered in future datasets, as they could make the landscape of irony and sarcasm detection more consistent.

6. Conclusion

In order to disambiguate the relationship between irony and sarcasm in NLP, we conducted a meta-analysis of the performance of different models on the seperate tasks of irony and sarcasm detection. The results of our analysis provide useful insights into how these two concepts should be treated in the future research.

Specifically, seperating the two tasks doesn't seem to be beneficial, as state-of-the-art models that perform well on one task, also perform well on the other. The overlap between the two concepts also seems to be large enough such that labeling them differently in datasets and/or detecting them separately isn't of much use.

Consequently, we propose a redefinition of the terms "irony" and "sarcasm" in NLP, either by further distinguishing between the two or by treating their detection as an even more related task.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

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.

Merriam-Webster. 2024. Merriam-webster dictionary.

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.

Silviu Oprea and Walid Magdy. 2020. iSarcasm: A dataset of intended sarcasm. In Dan Jurafssky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1279–1289, Online, July. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Rolandos Alexandros Potamias, Georgios Siolas, and Andreas-Georgios Stafylopatis. 2020. A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, 32(23):17309–17320, Dec.

D. Tomás, R. Ortega-Bueno, G. Zhang, et al. 2023. Transformer-based models for multimodal irony detection. *Journal of Ambient Intelligence and Humanized Computing*, 14:7399–7410.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 task 3: Irony detection in English tweets. In Marianna Apidianaki, Saif M. Mohammad, Jonathan May, Ekaterina Shutova, Steven Bethard, and Marine Carpuat, editors, *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana, June. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.