Contextualized Sarcasm Detection

Fangwei Gao* **Tianyi Lin** * **Yutian Zhao***
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{fangweig, tianyil2, yutianzh}@andrew.cmu.edu

Abstract

Sarcasm is a form of irony in which apparent praise conceals another, scornful meaning. Sarcasm is pervasive in social media such as Twitter and Facebook and can be highly disruptive to sentiment analysis and opinion mining systems. Hence, it is indispensable to develop effective models for sarcasm detection. In this project, we performed a thorough literature survey on sarcasm detection, implemented and examined four baseline models presented in iSarcasm (Oprea and Magdy, 2020), did a comprehensive error analysis on baseline models and implemented some improvements. By incorporating contextualized word embeddings and tweet context information into our sarcasm detection models, we are able to achieve state-of-the-art performance on iSarcasm dataset. Our code is published on Github[1].

1 Introduction

Sarcasm refers to the use of ironic utterances that mean the opposite of what you really want to say. People usually use sarcastic expressions to draw attention to some discrepancy between a description of the world they are putting forward and the way things actually were (Wilson, 2006). Due to its allegorical nature, sarcasm detection is a challenging task in natural language processing. Sarcasm is ubiquitous in social media and can be the biggest challenge in sentiment analysis and opinion mining (Liu, 2011). Therefore, an effective sarcasm detection model is beneficial to many opinion mining and sentiment analysis applications.

In this project, we implemented 4 baseline models (3CNN, LSTM, MIARN, and SIARN) described in iSarcasm (Oprea and Magdy, 2020) from scratch and achieved similar results to the results published in the original paper. We also did an error analysis from both model's and dataset's perspectives to identify some limitations of current approaches. Finally, we implemented several improvements based on the error analysis, including 1) pre-train with manual labeling dataset, 2) incorporate contextualized word embedding into our sarcasm detection task using BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2020) and BERTweet (Nguyen et al., 2020). 3) integrate tweet context information during training. Experiment results have shown that our approaches have better performance on author labeling dataset - iSarcasm.

2 Related Work

2.1 Sarcasm Datasets

There are mainly four methods used to label texts for sarcasm: distant supervision, manual labelling, a hybrid of both and labelling by original authors.

Distant supervision This is the most widely used method to label sarcasm datasets, especially for twitter texts. Texts are considered sarcastic if they meet some criteria, which most of them are simple and straightforward, such as whether certain tags are contained (Barbieri et al., 2014; Davidov et al., 2010; Ptáček et al., 2014; Riloff et al., 2013). Some are also considered positive if they are posted by special social media accounts such as "@spinozait" or "@LiveSpinoza" (Barbieri et al., 2014). Negative examples are mainly generated by randomly selecting twitters that don't meet the above criteria.

Manual Labeling Another commonly used method is to ask crowd workers to manually label the texts. (Abercrombie and Hovy, 2016; Filatova, 2012). This method could produce very different outcomes due to the various socio-cultural backgrounds of human annotators.

Author Labeling Recently, iSarcasm (Oprea

*Everyone Contributed Equally – Alphabetical order
[1]https://github.com/tianyiii/nnnlp_project

and Magdy, 2020) claims that the perceived sarcasm is very different from the original author's intended sarcasm and proposes to use an online survey to collect the authors' labels on their own tweets along with an explanation.

2.2 Sarcasm Detection Models

Previous works on sarcasm detection models mainly focus on the content of the text, while some also considers the context of the text and user profiling.

Content-based Most models view sarcasm detection problem as a binary classification problem and try to identify syntactic, lexical or pragmatic signals in text. Prosodic and spectral cues are used to detect sarcasm in spoken dialogue systems (Tepperman et al., 2006) while some use positive predicates, interjections and other linguistic features (Carvalho et al., 2009). Syntactic patterns are also used by several methods (Davidov et al., 2010; Tsur et al., 2010). Riloff (Riloff et al., 2013) tries to find positive sentiment along with negative situations. Some uses a combination of multiple features including implict, explicit, lexical and pragmatics (Joshi et al., 2015).

Most of the methods mention above use convolutional neural Network to extract location-invariant features that contain syntactic and semantic information and pass this representation vector through the final binary softmax layer (Hazarika et al., 2018).

Other methods use Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) to encode text and then add a binary softmax layer to output a probability distribution over labels. (Wu et al., 2018) adopt densely connected LSTM and (Yang et al., 2016) proposed Att-LSTM that adds an attention mechanism on top of the LSTM layer.

Context-based Researchers have claimed that extracting linguistic information from social medias such as twitter, blogs and discussion forums such as Reddit may be inaccurate and models may need additional clues to improve performance (Carvalho et al., 2009; Wallace et al., 2014). Most of the works focus on analyzing the historical posts of users (Rajadesingan et al., 2015; Zhang et al., 2016) and their attitudes and sentiments towards certain topics (Khattri et al., 2015). Certain works also explore context by learning user embeddings (Riloff et al., 2013) or personality representations from the input text (Poria et al., 2016).

Attention-based In the context of sarcasm detection, attention is often used as a method to learn sequence-level representations that selectively combine word-level representations based on their importance to the task. In some sequences, word-level linguistic incongruity can be viewed as a potential cause for sarcasm. SIARN (Single-Dimension Intra-Attention Network) and MIARN (Multi-Dimension Intra-Attention Network) use an intra-attention mechanism (Shen et al., 2017) to capture linguistic incongruity between words (Tay et al., 2018a). MIARN creates multiple intra-attention scores for each pair of words to capture multiple possible meanings of a word.

2.3 Contextualized Word Embeddings

Contextualized neural language models like BERT (Devlin et al., 2019) use deep neural networks to capture word interaction in its context. The multi-headed self-attention mechanism allows the sequence embedding, usually represented by the special [CLS] token's word representation, to learn important word interactions for many downstream tasks like information retrieval and sentiment analysis. This feature fits well with our sarcasm detection task.

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a large-scale languagem model pre-trained on English Wikipedia. Many models have achieved the state-of-the-art performance on their NLP tasks by fine-tuning BERT respectively. DistilBert leverages knowledge distillation during pre-training to reduce the model size of BERT while still maintain good performances on many downstream tasks. It's smaller model size makes it generally perform better on small dataset. BERTweet uses the same architecture as BERT's base model while it is pre-trained on a large-scale Twitter dataset. It achieves better performances on Tweet NLP tasks including text classification (Nguyen et al., 2020).

3 Dataset

We used the iSarcasm dataset introduced by (Oprea and Magdy, 2020) to evaluate our models. Among the 4484 tweets, we're only able to get 3828 texts (3072 train data, 756 test data) through Twitter API as the remaining tweets were removed by Twitter. Following the data preprocessing steps in (Oprea and Magdy, 2020; Tay et al., 2018b), texts are tokenized and truncated at 40. Texts with URLS(i.e.,

Model	Our Impl. (imbalanced)			Our Impl. (balanced)			Original Paper		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
3CNN	0.351	0.124	0.183	0.248	0.381	**0.301**	0.250	0.333	**0.286**
LSTM	0.205	0.686	0.315	0.212	0.810	**0.337**	0.217	0.747	**0.336**
SIARN	0.293	0.276	0.284	0.216	0.857	**0.345**	0.219	0.782	**0.342**
MIARN	0.276	0.410	0.330	0.223	0.771	**0.346**	0.236	0.793	**0.364**

Table 1: Experimental results on iSarcasm of our implemented models compared to the results published by (Oprea and Magdy, 2020). The iSarcasm paper didn't mention how they balance the dataset, so we trained our models on both imbalanced and balanced datasets. Since we're only able to get 3828 twitters out of 4484 tweets used by original paper through twitter API, the results are slightly different from the orginal paper.

containing 'http') are removed and mentions of other users in the text are replaced by '@USER'. Texts with less than 5 tokens after preprocessing are also removed.

Among the 3072 training data, 2543 of them are non-sarcastic and 529 are sarcastic. We used both imbalanced and balanced version (each sarcastic instance is replicated five times resulting in 2645 positive samples) to train the baseline models in section 4, and we only used balanced version to conduct experiment in section 5.

4 Baselines

4.1 Baseline Models

Following the experiment settings described in iSarcasm, we implemented the following 4 baseline models from scratch. We used the tweet tokenizer provided by nltk, and Glove twitter embeddings (Pennington et al., 2014) with 100 dimension to generate the input representations.

3CNN uses a convolutional neural network with three filters of sizes 3, 4, and 5 and 100 filters for each size (Oprea and Magdy, 2020; Hazarika et al., 2018). Each convolution layer is followed by a relu activation and a final dropout layer with probability 0.3.

LSTM encodes tweets with long-term short memory units (Hochreiter and Schmidhuber, 1997) using one hidden layer of dimension 100 (Oprea and Magdy, 2020). The LSTM layer is then followed by a dropout layer with probability 0.3 and a binary softmax layer to output a probability distribution over labels.

SIARN models word contrast and incongruity using an intra-attention mechanism (Shen et al., 2017). The largest attention score of each word is used in computing the sequence's intra-attentive representation. Instead of using a single attention score, **MIARN** uses an attention vector to model different views of a word. Attention score are then computed by feeding the vector into a single linear layer neural network.

What a fucking stupid country I have the misfortune to live in

Table 2: Visualization of normalized attention weights for a SIARN example. The intensity denotes the strength of the attention weight on the word.

4.2 Baseline Analysis

Among the four models, MIARN and SIARN achieved best F-score and outperformed LSTM and 3CNN on iSarcasm dataset.

However, after examining the attention generated by SIARN, we believe that the intra-attention mechanism used by SIARN and MIARN has some limitations. We noticed that most generated intra-attention representations are sparse, which means they attend to most tokens in the input sequences with only minor focus on linguistically incongruous words. See Table 2. There are several possible explanations for this observation. First, the attention training might not be complete due to the limited amount of tweets we used. Second, it is likely that a single layer intra-attention representation cannot fully incorporate all attentive information in all word pairs. Third, the max-pooling operation used in SIARN restricts each word's attention only to one other word in the sequence.

5 Experimental Approaches

We propose three primary approaches to improve sarcasm detection performance on iSarcasm dataset, outlined as follows:

5.1 Contextualized Word Representation

As discussed in the previous error analysis section, using sequence-level attention representations computed by a single attention layer may not be able to capture sufficient attentive information in a whole sequence. In order for the representation to capture more useful information, it is desired for the model to go deeper with word-level attention representation. BERT uses a multi-layer bidirectional Transformer encoder with multi-headed self-attention to learn contextualized word representations. It is a natural fit for our task and its special [CLS] and [SEP] tokens can be used efficiently in our work.

Our model implementation is straightforward. We first compute the sequence representation of an input sequence by applying BERT to it. The [CLS] token's representation of the last layer is used as the input sequence representation to a classifier. The classifier includes a linear layer and a dropout layer. It takes the sequence representation as input and compute logits for each class. In addition, due to the fact that iSarcasm is a relatively small tweet dataset, we also implemented DistilBERT and BERTweet to leverage these dataset features.

Although our model implementation is simple, we suffered heavily from fine-tuning instability that our training performance is highly unstable across different random seeds, since our dataset is small. To address this issue, we followed the advice of a recent paper (Mosbach et al., 2021) and adopted a new learning rate scheduler of two different phases. At the first 10% of all our fine-tuning steps, the learning rate gets increased linearly from 0 to its peak. In the next phase, the learning rate gets decreased from its peak to 0.

5.2 Pre-train with Manual Labeling Dataset

Since the training and testing datasets are all labelled by the original authors of the tweets in iSarcasm, the amount of data is very limited and the cost of manually label more data is high. In order to allow the sarcasm detection model to expose to more data and improve its performance, we collected a training corpus **mSarcasm**, 'm' stands for manual, for model pre-training.

mSarcasm includes datasets labelled by human annotators from various online sources [2] [3] [4] as we believe that trying to separate intended sarcasm with perceived sarcasm is essentially hard and unnecessary. The resulting mSarcasm corpus includes 173,539 training pairs with 111,152 non-sarcastic and 62,387 sarcastic tweets.

We pre-trained MIARN and SIARN, the top 2 baseline models, with newly collected mSarcasm corpus for one epoch. We call them MIARN$_{pretrain}$ and SIARN$_{pretrain}$. Then we fine-tuned the models with iSarcasm balanced training dataset. We believe that training the models on both author labeling and manual labeling datasets enables them to recognize both intended sarcasm and perceived sarcasm, and eventually improve their performance on the iSarcasm test data.

5.3 Integrate Tweet Context Information

Based on our error analysis, linguistic information extracted from a single sentence or a paragraph from a tweet may be inaccurate and the model still lack sufficient context information to detect sarcasm. Therefore, we would like to combine the raw content of the text with the **dialogue** information. After examining iSarcasm dataset, we found that looking only at one tweet at a time would be problematic as it may be a retweet or a reply to another tweet, and thus sarcasm is only detectable when looking at these tweets together in a dialogue setting. Therefore, we believe the model will benefit from including all the original tweets into the tweet text if its a reply or retweet.

We used Twitter API [5] to get all reference tweets of a tweet if it's a reply or retweet. Among 3072 training tweets in iSarcasm, 512 tweets have dialogue information. We concatenate dialogue information using [SEP] token and feed them into BERT during training.

6 Results and Analysis

We present quantitative results of our experiments in Table 3. We use Precision, Recall and F-score as our evaluation metrics following (Oprea and Magdy, 2020) to enable fair comparison. The results showed that our approaches outperformed baseline models and achieve state-of-the-art performance on iSarcasm dataset. In this section, we will discuss and perform some analysis about our experiment results.

[2] https://github.com/omidrohanian/irony_detection
[3] https://github.com/MirunaPislar/Sarcasm-Detection
[4] https://github.com/kbuschme/irony-detection
[5] https://developer.twitter.com/en/docs/twitter-api

Model	iSarcasm balanced		
	Precision	Recall	F-score
SIARN	0.216	0.857	0.345
MIARN	0.223	0.771	0.346
SIARN$_{pretrain}$	0.215	0.829	0.341
MIARN$_{pretrain}$	0.257	0.600	**0.360**
BERT	0.257	0.726	0.379
DistilBERT	0.205	0.847	0.330
BERTweet	**0.333**	0.444	**0.381**
BERT$_{context}$	0.270	0.670	**0.385**

Table 3: Complete list of experiment results conducting on iSarcasm balanced dataset.

6.1 Contextualized Word Representation

By leveraging contextualized word embedding into sarcasm detection, BERT and BERTweet all outperform the previous state-of-the-art model SIARN and MIARN by a large margin. These results show that capturing word interaction with its context information is a necessary part for improving a model's ability to detect sarcasm in text data. Particularly, BERTweet outperforms other models a lot in terms of accuracy. We think its difference in pre-training data source helps it to model word meaning better in the context of tweet data, which leads to its good performance on iSarcasm dataset.

6.2 Pre-training on mSarcasm

Using newly collected mSarcasm corpus to pre-train MIARN model improved its F1 score from 0.346 to 0.360. However, we noticed that it didn't change the performance of SIARN model. We believe this is due to the complex architecture of MIARN model, so that the model hasn't been trained fully on limited tweets in iSarcasm, and thus can benefit from pre-training.

We also noticed that pre-training helped SIARN and MIARN converge faster when trained/fine-tuned on iSarcasm dataset. Both models yielded the best results within 5 epochs with pre-training, while they converged to the optimal results around 30 epochs when trained on iSarcasm directly.

However, incorporating mSarcasm deteriorate performance of BERT-based models. We also did similar experiment to first fine-tune BERTweet with mSarcasm manual labeling dataset, and then we used fine-tuned BERTweet to initialize weights. Although BERTweet achieved a very good score on mSarcasm with F-score = 0.964, it performed poorly and didn't even converge when we fine-tuned it on iSarcasm dataset. This also indicates that BERT-based models could potentially distinguish between intended sarcasm and perceived sarcasm, so knowledge learned from perceived sarcasm can not be transferred to detect intended sarcasm directly.

6.3 Incorporating Dialogue Context

As shown in Table 3, BERT$_{context}$ that incorporates dialogue information is able to improve BERT F-score from 0.379 to 0.385. This result proves our hypothesis that some sarcasm tweets are only detectable when looking at them in a dialogue setting. We believe that the improvement will be more significant if we could get more context information, such as blog information if a tweet mentions another Twitter blog.

7 Conclusion

In this work, we propose several improvements over sarcasm detection models on the author labeling dataset - iSarcasm, including incorporating contextualized word embedding, pre-training with manual labeling dataset, and integrating tweet context information during training. The results show that our approaches achieve better performance compared to our baselines. Future work includes using distant supervision to collect data that reflect author intention, and analyzing a user's historical posts and socio-cultural traits to incorporate user embedding during training.

References

Gavin Abercrombie and Dirk Hovy. 2016. Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations. In *Proceedings of the ACL 2016 student research workshop*, pages 107–113.

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2014. Italian irony detection in twitter: a first approach. In *The First Italian Conference on Computational Linguistics CLiC-it*, volume 28.

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*, pages 53–56.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcasm in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116.

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

Elena Filatova. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Lrec*, pages 392–398. Citeseer.

Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. Cascade: Contextual sarcasm detection in online discussion forums. *arXiv preprint arXiv:1805.06413*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762.

Anupam Khattri, Aditya Joshi, Pushpak Bhattacharyya, and Mark Carman. 2015. Your sentiment precedes you: Using an author's historical tweets to predict sarcasm. In *Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 25–30.

Bing Liu. 2011. Opinion mining and sentiment analysis. In *Web Data Mining*, pages 459–526. Springer.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines.

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.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. *arXiv preprint arXiv:1610.08815*.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, pages 213–223.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the eighth ACM international conference on web search and data mining*, pages 97–106.

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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2017. Disan: Directional self-attention network for rnn/cnn-free language understanding.

Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018a. Reasoning with sarcasm by reading in-between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1010–1020, Melbourne, Australia. Association for Computational Linguistics.

Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. 2018b. Reasoning with sarcasm by reading in-between. *arXiv preprint arXiv:1805.02856*.

Joseph Tepperman, David Traum, and Shrikanth Narayanan. 2006. " yeah right": Sarcasm recognition for spoken dialogue systems. In *Ninth international conference on spoken language processing*.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 4.

Byron C Wallace, Laura Kertz, Eugene Charniak, et al. 2014. Humans require context to infer ironic intent (so computers probably do, too). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 512–516.

Deirdre Wilson. 2006. The pragmatics of verbal irony: Echo or pretence? *Lingua*, 116(10):1722–1743.

Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. 2018. Thu_ngn at semeval-2018 task 3: Tweet irony detection with densely connected lstm and multi-task learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 51–56.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: technical papers*, pages 2449–2460.