BERT's The Word: Sarcasm Target Detection using BERT

Pradeesh Parameswaran, Andrew Trotman, Veronica Liesaputra, David Eyers
Department of Computer Science
University of Otago
New Zealand
[pradeesh,andrew,veronica,dme]@cs.otago.ac.nz

Abstract

In 2019, the Australasian Language Technology Association (ALTA) organised a shared task to detect the target of sarcastic comments posted on social media. However, there were no winners as it proved to be a difficult task. In this work, we revisit the task posted by ALTA using transformers—specifically BERT—given the current success of the transformer-based model in various NLP tasks. We conducted our experiments on two BERT models (TD-BERT and BERT-AEN). We evaluated our model on the data set provided by ALTA ('Reddit') and two additional data sets: 'book snippets' and 'Tweets'. Our results show that our proposed method achieves a 15.2% improvement from the current state-of-the-art system on the Reddit data set and a 4% improvement on Tweets.

1 Introduction

Sarcasm is a remark made by a certain person to ridicule or hurt another person's feelings (Cheang and Pell, 2008). A unique property of sarcasm lies in the way words are used. The result digresses from the conventional word order and alters the meaning of the whole sentence (Attardo et al., 2003). This very aspect also makes it very challenging to detect in a text. There has been a large number of studies that looked at automating sarcasm detection (Eke et al., 2020; Joshi et al., 2017) however there is less work done in identifying and extracting the target of sarcasm from the text.

The problem of sarcasm target detection was originally coined by Joshi et al. (2016a). The *target of sarcasm* is defined as an entity or a situation that is being ridiculed in a sarcastic text. The task of sarcasm target identification is to extract the subset of words that indicate the target of ridicule for a given sarcastic sentence. Identifying the target of ridicule can improve the detection of cyber-bullying and hate speech targeted towards minority communities such as people of colour, the LGBTQ+ community and others (Oliva et al., 2021; Hylton, 2018). However, this task is particularly challenging because of the following factors:

- **Multiple targets**—A sarcastic sentence may contain multiple targets. For instance in the following sentence, "*James is as good at cooking as Guy Feiri is at avoiding controversy*", the targets are both "*James*" and "*Guy Feiri*".

- **Lack of targets**—The target of sarcasm may not be present in the given sentence. For example in the sentence, "*I guess the kumara loves kayaking*", the speaker makes a sarcastic remark but the target of ridicule is unclear. When the sarcasm target does is not present or it is unclear, it is marked as OUTSIDE.

There have been various attempts to improve the performance of sarcasm target detection (Patro et al., 2019; Molla and Joshi, 2019; Bölücü and Can, 2020; Parameswaran et al., 2021) such as through the use of deep-learning models and rule-based methods. Given the successes of transformers, particularly BERT (Devlin et al., 2019), in NLP tasks such as Aspect-Based Sentiment Analysis (ABSA) (Sun et al., 2019a), and summarisation (Miller, 2019), we hypothesise that BERT-like models may also be good at this task.

Our experiments show that BERT models outperform the current state-of-the-art system on our Reddit data by 23.4% and give a 3% increase on our Tweets data.

2 Related Work

Sarcasm detection (i.e., distinguishing sarcastic texts from non-sarcastic texts) is widely studied in computational linguistics. Eke et al. (2020); Joshi et al. (2017) have presented a comprehensive overview in this field. To summarise, there are several approaches: semi-supervised learning (Bamman and Smith, 2015; Bharti et al., 2015; Ling and Klinger, 2016; Ghosh and Muresan, 2018), deep learning (Ghosh and Veale, 2016; Agrawal and An, 2018; Hazarika et al., 2018; Martini et al., 2018; Liu et al., 2019), and lately, with the advancement of transformers, researchers have used transformers to distinguish sarcastic texts from non-sarcastic texts (Baruah et al., 2020; Avvaru et al., 2020; Potamias et al., 2020).

Little work has been done to detect the target of sarcasm—in spite of the Australasian Language Technology Association (ALTA) organising a shared task challenge to encourage researchers to tackle this problem (Molla and Joshi, 2019). The approaches taken in prior work include a rule-based system that looks at Part-of-Speech (PoS) (Joshi et al., 2016b), deep learning (Patro et al., 2019), and an ensemble of machine learning and deep learning classifiers (Parameswaran et al., 2021). To the best of our knowledge, there is no research exploring the use of BERT models for sarcasm target detection, but we note that Parameswaran et al. (2021) used embeddings from BERT, but not the transformer.

BERT has shown success in Aspect-Extraction (AE) within ABSA tasks (Xu et al., 2019; Hoang et al., 2019). Our task is similar to Aspect-Extraction, but in our task, the targets may be absent from the given text, which makes it challenging. We consider two models from the ABSA literature for our experiments: TD-BERT (Gao et al., 2019) and BERT-AEN (Song et al., 2019). Initially our choice was guided by the fact that BERT-AEN works well with a smaller data set (Gao et al., 2019) and we chose TD-BERT as our second option as we have noticed similiar performance to BERT-AEN with a much simpler architecture by just extending it to include the aspect. Our choice to use these models is further motivated by the availability of a public repository[1] with standard implementations using *PyTorch*[2] (and therefore ease of reproducibility of our experiments).

[1]https://github.com/songyouwei/ABSA-PyTorch/
[2]https://pytorch.org/

	Tweets	*Books*	*Reddit*
Sentences	224	506	950
Avg. sentence length	13.06	28.47	25.30
Avg. target length	2.08	1.6	2.8
% OUTSIDE	10%	5%	35%

Table 1: Statistics of data sets

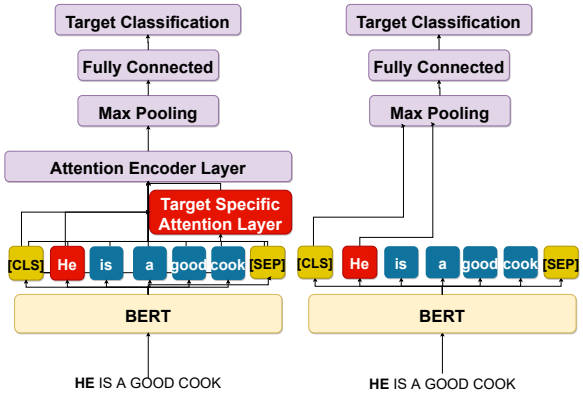

Figure 1: The architecture of BERT-AEN (Song et al., 2019) (left) and TD-BERT (Gao et al., 2019) (right)

We also note that Mukherjee et al. (2021) used the ABSA-PyTorch repository as the basis for reproducing the results of ABSA approaches.

3 Data Set

We consider the data sets released by Joshi et al. (2016b) and Molla and Joshi (2019). The sets consist of three different kinds of data: Tweets (*Tweets*), book snippets (*Books*) and also Reddit posts (*Reddit*). Table 1 shows the details of the collections.

4 Methodology

When predicting the target of sarcasm, like others (Patro et al. (2019) and Parameswaran et al. (2021)), we formulated the problem as a sequence labelling problem. We first represent a sarcastic sentence (S) as a sequence of words $\{w_1, w_2, \ldots, w_N\}$. We then append each word with a label indicating if it is a potential target. Consider the following example, "*He is a good cook*" with '*He*' as the potential target. The sentence is represented as $\{$*'He'*, *'is'*, *'a'*, *'good'*, *'cook'*$\}$ and its label sequence is $\{$*'He'*$_T$, *'is'*$_\emptyset$, *'a'*$_\emptyset$, *'good'*$_\emptyset$, *'cook'*$_\emptyset\}$, where T is a potential target and $\emptyset$ is not. We feed both of these sequences as training input for our two BERT models (TD-BERT and BERT-AEN).

Model	*Tweets*	*Books*	*Reddit*
Baseline 1 (Patro et al., 2019)	0.831 ± 0.156	0.865 ± 0.188	0.623 ± 0.261
Baseline 2 (Parameswaran et al., 2021)	0.860 ± 0.165	0.879 ± 0.194	0.715 ± 0.260
TD-BERT	0.849 ± 0.123	$\mathbf{0.881} \pm 0.195$	0.663 ± 0.245
BERT-AEN	0.848 ± 0.102	0.864 ± 0.172	0.689 ± 0.274
TD-BERT (*PT*)	$\mathbf{0.891} \pm 0.153$	NA †	$\mathbf{0.824} \pm 0.303$
BERT-AEN (*PT*)	0.880 ± 0.183	NA †	0.785 ± 0.299

Table 2: Results of our experiments. The figures in each case indicate the mean DICE score and standard deviation. *PT* denotes that the model is further trained to understand the nuances of the data set. † denotes that the scores for non-*PT* and *PT* models are the same as we did not further train BERT on *Books*.

Since we are classifying whether each word in a sentence is a potential target of sarcasm, the first word of the sequence is appended with a unique `[CLS]` token which is used by BERT for classification tasks. As shown in Figure 1, in order to train the model, we transform the given sentence (S) into `[CLS]` + S + `[SEP]` and `[CLS]` + w_k + `[SEP]` along with the label, w_k, where k is $\{1 \dots N\}$. As there can be multiple potential target terms, we introduce a max-pooling operation to the two BERT models. This takes into consideration which candidate targets are the best before it gets fed into the fully connected layer. Finally, we use a softmax layer in order to classify whether the current word is a potential target of sarcasm or not. We use $\text{BERT}_{\text{Base}}$ (Devlin et al., 2019) as our pre-trained model.

We briefly explain the architecture of the TD-BERT and BERT-AEN models below:-

- **BERT-AEN** (Song et al., 2019)—This model uses an attention encoder network to model the semantic interaction between the whole sentence and the potential target. The Target Specific Attention Layer is introduced so that it can compute the hidden states of the input embedding. The attentional encoder layer has two submodules: multi-head attention (MHA) and point-wise convolution transformation (PCT). The MHA performs multiple attention functionality that provides introspective context words modelling and perceptive target word modelling. According to Song et al. (2019), this is a lightweight solution as opposed to using LSTM. Then, PCT transforms the contextual information from MHA by incorporating context-perceptive target words. Additionally, BERT-AEN uses label smoothing regularisation (LSR) in the loss function. LSR reduces overfitting by replacing the 0 and 1 targets for the classifier with smoothed values (such as 0.1 and 0.9, respectively). This works well in our situation, where we have a limited amount of data.

- **TD-BERT** (Gao et al., 2019)—TD-BERT's architecture closely resembles that of BERT. The key difference is that TD-BERT incorporates the potential target information in its classification input, as described above.

Given the small number of sentences in our data sets, and the domain specific language used in *Reddit* and *Tweets*, we initially trained $\text{BERT}_{\text{Base}}$ to additionally understand the nuances of language use in those domains (Sun et al., 2019b). To do this we sampled 150,000 posts from Khodak's Reddit data set (Khodak et al., 2017) for *Reddit* and 100,000 tweets from The Edinburgh Twitter Corpus (Petrović et al., 2010) for *Tweets*. We further pre-trained $\text{BERT}_{\text{Base}}$ as a Mask Language Modelling task. We followed the recommendation of (Devlin et al., 2019), by masking 15% of all input tokens randomly. Additionally, we took the necessary steps to ensure that the sentences found in *Reddit* and *Tweets* were removed from Khodak's Reddit and the Edinburgh Twitter Corpus before training. We did no additional training for *Books* because $\text{BERT}_{\text{Base}}$ has already been trained on such content (Devlin et al., 2019).

We reserved 10% of our training set for the purpose of fine-tuning parameters. The best parameters we found were a batch size of 32, a maximum sequence length of 128, the maximum predictions per sequence being 20, and a learning rate of 10^{-5}.

Once we had trained the models for *Reddit* and *Tweets*, we then fine-tuned both of our BERT models to each of our three data sets using the training data provided in those data sets. We set the number of epochs to 3 and the *learning_rate* to be 10^{-5}

following the recommendation from Devlin et al. (2019).

5 Experimental Setup

We ran our experiments on an Intel Xeon E5-2690 v3 @ 2.00 GHz CPU with an NVIDIA Tesla T4 (CUDA Version 11.2, Driver Version 460.73) running on Debian 10 (Buster). We forked commit `9acab7e` of ABSA-PyTorch and modified it to suit our task. Our source code can be found on our GitHub page.[3]

We consider the current state-of-the-art models from Patro et al. (2019) and Parameswaran et al. (2021) as baselines for this task, that we called Baseline 1 and Baseline 2, respectively. We implemented the approaches of each author and compared our results to theirs. A one-way ANOVA showed no statistically significant difference at the 0.05 level, providing confidence in our implementations of their approaches.

We use DICE score to measure the accuracy as it has been used in past works (Joshi et al., 2016a; Molla and Joshi, 2019). All the results reported used five-fold cross-validation.

6 Results

We report our results in Table 2. It is not surprising that training BERT improves results for *Reddit* and *Tweets* as the model has learned the nuances of language used on those platforms (Sun et al., 2019b). From our experimental results, training TD-BERT gives a 15.2% improvement on state-of-the-art for *Reddit*, but only a modest improvement of 4% for *Tweets* and 0.22% for *Books*.

Surprising to us, TD-BERT performs best in all our tasks. We believe that the simple method of just incorporating a target's position helped the model to better understand the context of the sentences. Although the multiple attention mechanisms in BERT-AEN could be expected to outperform TD-BERT, it is unclear why this is not the case in our experimental results. One possible explanation is that the data set is small and the model has learned more noise. We leave for future work the exploration using larger data sets.

6.1 Evaluation on Kaggle

In addition to evaluating our models on the three data sets, we ran our best performing model (TD-BERT (*PT*)) on the data from the ALTA 2019 Shared Task, as seen on Kaggle. This allowed us to examine the generalisability of our solution *in the wild*. Table 3 presents our results and the results from previously published runs. A one-way ANOVA-test of our model with Baseline 1 and Baseline 2 did not find any statistically significant difference at the $p < 0.05$ level. However, our approach beats all the participants' runs (*Powers* and *Orangutan*) and the two baselines provided by the Shared Task (*Always* OUTSIDE, which always outputs 'no target', and *Pronouns*, which extracts and outputs the pronouns) at the $p < 0.05$ level.

System	Public	Private
Baseline 1 (Patro et al., 2019)	0.466	0.514
Baseline 2 (Parameswaran et al., 2021)	0.493	0.548
Always OUTSIDE	0.367†	0.349†
Powers	0.386†	0.333†
Orangutan	0.371†	0.292†
Pronouns	0.209†	0.225†
Ours (TD-BERT (*PT*))	**0.501**	**0.562**

Table 3: Evaluation on Kaggle Public and Private portions of the data set. † denotes a method that was included within the 2019 ALTA Shared Task Challenge

We further investigated the Kaggle score as our model's DICE score is much lower than the DICE score that we obtained in the other three datasets. First, we validated the scores in Table 2 by uploading our test portion to a private Kaggle contest and evaluating our run. The Kaggle score matched that in the table, giving confidence that our implementation of DICE is correct.

Next, we augmented our sentences with the subreddit information from Khodak et al. (2017) and compared that to the annotated public portion of *Reddit*. We observe that 23% of the private portion's subreddits are not in the public portion. We hypothesise that the model has learned the nuances of the subreddits it has seen, but cannot generalise this across all subreddits. However, we do not have the ground truth, so cannot form any solid conclusions.

6.2 Computational Costs

Figure 2 illustrates the comparisons of our chosen models' run-time and evaluation time on all three of the data sets. We can see that the much simpler TD-BERT performs faster than BERT-AEN in all the cases. The training and evaluation time for

[3] https://github.com/prasys/ABSA-PyTorch/

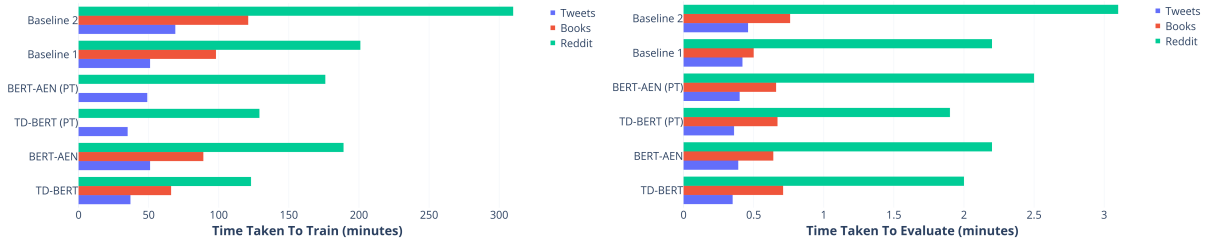

Figure 2: TD-BERT and BERT-AEN training time (left) and run time (right) comparison

non *PT* models are very similar to the *PT* ones. However, it is worth noting that training further on the *Reddit* data set took 336 minutes, and for *Tweets* took 240 minutes. We believe that the performance gain for both data sets easily justifies this modest training time. Hao et al. (2019) suggested that running more epochs can improve performance but we leave this for future work.

6.3 Failure Analysis

Compared to the untrained $\text{BERT}_{\text{Base}}$, we believe that our trained $\text{BERT}_{\text{Base}}$ can better understand the language used in various subreddits, where there are often novel words being coined. Consider the following example from the training set, "*Yeah, an ice cream is so much less creative than a* ***pokeball with eyes***" (target in bold). The further trained model predicted a partially correct answer of *"pokeball"* but the out of the box model misclassified this sentence, returning OUTSIDE.

However, there are instances where we did not obtain the correct target, regardless of the BERT model or any additional training and fine-tuning. For example, *"Yeah 0i have an i5 520m and Intel HD and you know, it really bugs the hell out of me when my fps goes below 20 like come on"*. The annotators mark it as OUTSIDE but both of our models predicted *"I"*, we believe the answer to be *"Intel HD"* as well as *"i5 520m"*.

In *Reddit*, the standard deviation of the DICE scores is higher than in the other data sets. This lends further evidence to our hypothesis that domain (subreddit) specific language is learned in training, and is not easily generalised. Patro et al. (2019) has demonstrated that a PoS tagger can help improve the quality of a sarcasm target detector, and we believe it might help here too. We leave the exploration of this for future work.

7 Conclusion

We presented our approach to sarcasm target detection. We used two different publicly available BERT models: TD-BERT and BERT-AEN, and fine-tuned them to the task using extra examples of data from the domains we explore. Finally, we evaluated our models on three publicly available data sets: *Tweets*, *Books*, and *Reddit*. Our empirical results show that this approach outperforms the current state-of-the-art on all three data sets.

Despite setting a strong baseline, we believe that there remains plenty of room for further work in this area. Firstly, we conducted our experiments on a small data set, therefore our proposed methodology needs to be tested when applied to a larger data set. Secondly, the use of user profiles, user history, context, and so on, might improve performance for *Reddit* and *Tweets* as detecting sarcasm is a difficult task and it requires more than content alone. Some users are more prone to sarcastic quips than others, and that could be mined from a person's past posts (Marwick and Boyd, 2011).

Acknowledgements

This material is based upon work supported by the Google Cloud Research Credits program with the award GCP19980904.

References

Ameeta Agrawal and Aijun An. 2018. Affective representations for sarcasm detection. In *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*, pages 1029–1032.

Salvatore Attardo, Jodi Eisterhold, Jennifer Hay, and Isabella Poggi. 2003. Multimodal markers of irony and sarcasm.

Adithya Avvaru, Sanath Vobilisetty, and Radhika Mamidi. 2020. Detecting sarcasm in conversation context using transformer-based models. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 98–103.

David Bamman and Noah A. Smith. 2015. Contextualized sarcasm detection on twitter. In *Proceedings of the 9th International Conference on Web and Social Media, ICWSM 2015*, pages 574–577.

Arup Baruah, Kaushik Das, Ferdous Barbhuiya, and Kuntal Dey. 2020. Context-aware sarcasm detection using bert. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 83–87.

Santosh Kumar Bharti, Korra Sathya Babu, and Sanjay Kumar Jena. 2015. Parsing-based sarcasm sentiment recognition in Twitter data. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015*, pages 1373–1380.

Necva Bölücü and Burcu Can. 2020. Sarcasm target identification with lstm networks. In *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE.

Henry S Cheang and Marc D Pell. 2008. The sound of sarcasm. *Speech communication*, 50(5):366–381.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 4171–4186.

Christopher Ifeanyi Eke, Azah Anir Norman, Liyana Shuib, and Henry Friday Nweke. 2020. Sarcasm identification in textual data: systematic review, research challenges and open directions. *Artificial Intelligence Review*, 53(6):4215–4258.

Zhengjie Gao, Ao Feng, Xinyu Song, and Xi Wu. 2019. Target-dependent sentiment classification with bert. *IEEE Access*, 7:154290–154299.

Aniruddha Ghosh and Dr. 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.

Debanjan Ghosh and Smaranda Muresan. 2018. "With 1 follower I must be AWESOME :P". Exploring the role of irony markers in irony recognition. *Icwsm*, (Icwsm):588–591.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of bert. *arXiv preprint arXiv:1908.05620*.

Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. CASCADE: Contextual Sarcasm Detection in Online Discussion Forums. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1837–1848.

Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. 2019. Aspect-based sentiment analysis using bert. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 187–196.

Kevin Hylton. 2018. I'm not joking! the strategic use of humour in stories of racism. *Ethnicities*, 18(3):327–343.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys*, 50(5):1–22.

Aditya Joshi, Pranav Goel, Pushpak Bhattacharyya, and Mark Carman. 2016a. Automatic Identification of Sarcasm Target: An Introductory Approach.

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, Mark Carman, Meghna Singh, Jaya Saraswati, and Rajita Shukla. 2016b. How challenging is sarcasm versus irony classification?: A study with a dataset from English literature. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 123–127.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579*.

Jennifer Ling and Roman Klinger. 2016. An empirical, quantitative analysis of the differences between sarcasm and Irony. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9989 LNCS:203–216.

Liyuan Liu, Jennifer Lewis Priestley, Yiyun Zhou, Herman E. Ray, and Meng Han. 2019. A2Text-Net: A Novel Deep Neural Network for Sarcasm Detection. In *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*, pages 118–126.

Andrianarisoa Tojo Martini, Makhmudov Farrukh, and Hongwei Ge. 2018. Recognition of ironic sentences in Twitter using attention-based LSTM. *International Journal of Advanced Computer Science and Applications*, 9(8):7–11.

Alice E Marwick and Danah Boyd. 2011. I tweet honestly, i tweet passionately: Twitter users, context collapse, and the imagined audience. *New media & society*, 13(1):114–133.

Derek Miller. 2019. Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*.

Diego Molla and Aditya Joshi. 2019. Overview of the 2019 ALTA Shared Task : Sarcasm Target Identification. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 192–196.

Rajdeep Mukherjee, Shreyas Shetty, Subrata Chattopadhyay, Subhadeep Maji, Samik Datta, and Pawan Goyal. 2021. Reproducibility, replicability and beyond: Assessing production readiness of aspect based sentiment analysis in the wild. *arXiv preprint arXiv:2101.09449*.

Thiago Dias Oliva, Dennys Marcelo Antonialli, and Alessandra Gomes. 2021. Fighting hate speech, silencing drag queens? artificial intelligence in content moderation and risks to lgbtq voices online. *Sexuality & Culture*, 25(2):700–732.

Pradeesh Parameswaran, Andrew Trotman, Veronica Liesaputra, and David Eyers. 2021. Detecting the target of sarcasm is hard: Really?? *Information Processing & Management*, 58(4):102599.

Jasabanta Patro, Srijan Bansal, and Animesh Mukherjee. 2019. A deep-learning framework to detect sarcasm targets. 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 6335–6341.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. The edinburgh twitter corpus. In *Proceedings of the NAACL HLT 2010 workshop on computational linguistics in a world of social media*, pages 25–26.

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.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *arXiv preprint arXiv:1902.09314*.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019a. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019b. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*.