Sarcasm Detection Models are *So* Easy to Improve! /s

Diana Abagyan
dabagyan@uw.edu

Libby Merchant
emercha@uw.edu

Emma McKibbin
echm@uw.edu

Jade Phoreman
jphore@uw.edu

Catherine Ball
catball@uw.edu

Abstract

In the present study, we build a simple sarcasm detection system using a BERT model fine-tuned on the Self-Annotated Reddit Corpus (SARC). We attempt to improve this model by leveraging context, other pre-trained embedding models, and ensembling techniques. We also introduce the Self-Annotated Neurodivergent (SAND) Corpus, targeting tone indicator usage in online neurodivergent communities, and adapt our model for classification on this new dataset. We find that the best performing systems for either task are specialized to a single task, while a more generalized system may perform below state-of-the-art on both tasks.

1 Introduction

Sarcasm detection is a notoriously difficult task, both for the detector and for the task developer. The Self-Annotated Reddit Corpus (SARC) (Khodak et al., 2018) leverages the use of explicit tone indicators in online communities to more reliably annotate their task data. These tone indicators are popular in neurodiverse communities, in part because neurodivergent people are less confident in their interpretation of non-literal language online. We train our initial system on SARC, which pulls data from general populations on Reddit. Then we adapt our system on a self-collected task built from posts in neurodiverse Reddit communities.

Previous studies make use of a variety of models for sarcasm detection, such as SVMs (Tungthamthiti et al., 2016), CNNs (Hazarika et al., 2018), and ensemble approaches adding LSTMs and MLPs (Lemmens et al., 2020). In general, the literature shows a shift from using GloVe to using BERT (Devlin et al., 2019) for embeddings, so we begin with an initial system using BERT as our classifier. We then revise our system in two ways: first, by augmenting our input with the previous comment as context; second, by using RoBERTa (Liu et al., 2019) as our embedding model in place of BERT. We find that the model with both of these revisions performs best on our development set.

2 Related Work

Automatic sarcasm detection tasks have historically posed serious challenges as they are strongly dependent upon context. A variety of model architectures have been proposed, such as Support Vector Machines (Tungthamthiti et al., 2016), and ensemble approaches (Lemmens et al., 2020). However, performance varies depending on data source and conversational context provided. Baruah et al. (2020) shows that contextual improvements for a sarcasm detection system on Twitter data may not improve a system processing Reddit data. Models that perform well on Reddit data, such as CASCADE in Hazarika et al. (2018), may make use of non-linguistic data, like user behavior, to better predict sarcasm. However, strategies like these not only require user data that may not be published with every dataset but also run the risk of introducing unethical bias into the system.

Further, the task of sarcasm detection has largely relied on labeled datasets created by third-person annotators. But this task is difficult not just for models but for humans, as well. As Khodak et al. (2018) reports, when human evaluators were presented with two statements and asked to identify which of the two was sarcastic, the average human F1-score was 0.816. Therefore, when developing sarcasm detection datasets, labeling data as sarcastic or non-sarcastic is non-trivial, as it relies strongly upon subjective human judgement.

Of note, neurodivergent (ND) individuals have been shown to process sarcasm in different ways from those who are typically developing: Zalla et al. (2014) indicated that Autism Spectrum Disorder (ASD) individuals are less likely to respond to social stereotypes as a marker of sarcasm, while Ludlow et al. (2017) suggested that children with

Attention-Deficit Hyperactivity Disorder (ADHD) are less likely to comprehend paradoxical sarcasm in particular.

Previous studies have not reached a consensus as to whether ND individuals show a deficit in detecting ironic language when overall language ability is controlled for (Kalandadze et al., 2018). In an eye-tracking study on ASD and typically-developing participants, Au-Yeung et al. (2015) found similar accuracy in sarcasm detection between the two groups; however, the ASD group took more time to read and process the statements, suggesting that these individuals have lower confidence in their ability to detect irony. For some in these groups, the task of sarcasm detection poses its own challenge.

In response to this, tone indicators have emerged in neurodiverse communities online as a way to compensate for the unique difficulties of communication in online spaces (Febiana Christanti et al., 2022). Online text does not offer the paralinguistic cues such as prosody, facial expressions, and body language that help disambiguate sarcasm in face-to-face interactions.

Additionally, this linguistic innovation has been explored as a way to circumnavigate the challenges of developing labeled corpora for automated sarcasm detection. The Self-Annotated Reddit Corpus (SARC) put forth by Khodak et al. (2018) proposed a way to extract self-annotated sarcasm datasets through tone indicators in social media posts. This is particularly salient for the purpose of disambiguating ironic language in neurodiverse spaces, due to the differing reactions to ironic language within these groups.

3 Task Description

We develop a sarcasm detection model first on our primary task, which uses the SARC datset. Then, we adapt our model to our self-collected adaptation task, which differs from SARC primarily in the population producing the self-annotated posts. In SARC, the population is general; in our dataset, the population is restricted to communities with high concentrations of ND members.

3.1 Primary Task

Our primary task is sarcasm detection using a model trained on the SARC dataset.[1] This task is characterized by the following dimensions:

- **Affect type**: emotion (sarcasm)
- **Recognition type**: classification
- **Genre**: Reddit posts
- **Target**: N/A
- **Modality**: text
- **Language**: English

The SARC dataset was collected from Reddit posts and comprises 533M comments total, with 1.3M sarcastic comments. All comments include data about the author, topic, and context. The data is self-annotated in the sense that users labeled their own sarcastic comments using the tone indicator */s*.

We utilize the balanced portion of the dataset during model development. The sizes of each partition used in training, development, and testing are described in Table 1. Note that we created a development set from 10% of the provided balanced training set.

Partition	Count
Original Balanced Training	257,082
Training	231,374
Development	25,708
Balanced Testing	64,666

Table 1: The size of each partition from the SARC dataset used in our study. We train on "Training without Dev," do intermediate evaluation on "Development," and finally evaluate on "Balanced Testing."

In the SARC data, there is some level of noise associated with self-annotation using tone indicators; some sarcastic statements may not be labeled as such, and those that are labeled may only be as such because they are especially ambiguous (Khodak et al., 2018). The first source of noise can be mitigated in our own data collection by targeting only those posts labeled as serious (represented by */srs*), but this exacerbates the second source of noise.

3.2 Adaptation Task

Our adaptation task is another sarcasm detection task with similar dimensions as our primary task. However, our adaptation data is sourced from online communities that we believe to have significantly higher proportions of ND users.

[1] SARC data: https://nlp.cs.princeton.edu/old/SARC/2.0/

We collected text posts from subreddits such as r/Neurodivergent, r/neurodiversity, and r/autism, where self-labeling sarcasm with the */s* tone indicator is somewhat common. There does not seem to be any literature on using computational models to perform sarcasm detection among ND populations specifically, so instead we refer to adjacent literature for guidance, such as Au-Yeung et al. (2015) and Febiana Christanti et al. (2022). Our final dataset is known as the SAND (**S**elf-**A**nnotated **N**euro**D**ivergent) Corpus for sarcasm detection.

3.2.1 Data Collection

Our adaptation task necessitated the creation of a new dataset primarily authored and labeled by ND netizens. Reddit offers a convenient way to narrow the search for this data due to its prominence of neurodiversity-related forums, or *subreddits*. To target users belonging to these communities, we limited our data sources to relevant subreddits, listed in Appendix A.

The final dataset merged data points from an initial scrape and data retrieved from the Pushshift Reddit Dataset (16.8M instances). There was no overlap between the two given the different time frames in which they were collected. We found all instances of the "/s" tone indicator and balanced those instances with non-sarcastic instances, including those that use "/srs", as well as unmarked instances from authors who had used "/s" elsewhere in the data. Our final dataset has over 430K instances. A breakdown of the partition sizes can be seen in Table 2

Partition	Count
Training	337,782
Development	48,236
Testing	48,186

Table 2: The size of each partition in our SAND corpus, collected for our study. The number of sarcastic and non-sarcastic instances are roughly balanced in each partition.

4 System Overview

In total, we developed 7 iterations of our sarcasm detection system. In D2, we created just our initial system: a BERT classifier fine-tuned on the SARC data, without considering context. In D3, we developed 3 additional systems, all fine-tuned on SARC data. The first was a no-context RoBERTa classifier. Then, we appended previous context as input in order to fine-tune new BERT and RoBERTa systems. In D4, we developed 2 systems. To try and improve our primary system, we created an ensemble model that used our no-context, SARC-fine-tuned BERT and RoBERTa models as base models. To better accommodate our adaptation task, we further fine-tuned our no-context RoBERTa classifier on our SAND corpus data.

Our final system, developed after the submission of D4, is an ensemble model which has as its base models our 2 no-context RoBERTa models—one fine-tuned only on SARC and the other fine-tuned further on SAND.

BERT Our initial system architecture was adapted from (Baruah et al., 2020) and comprises a pre-trained BERT model with a classification layer. For this, we used BertForSequenceClassification as accessed through the transformers library. We then fine-tuned this model for sarcasm detection before performing classification on our primary and adaptation tasks.

RoBERTa Liu et al. (2019) reproduced the original BERT experiment, paying extra attention to the effects of the model's hyperparameters and the training task. Most notably, their model, RoBERTa, underwent significantly more pre-training than BERT, resulting in a model that outperforms BERT on at least one sentiment analysis task. For this reason, we test whether using RoBERTa in place of BERT for embeddings and classification can improve our system's performance. We use the RobertaForSequenceClassification class in HuggingFace, which also adds a linear layer for classification after the RoBERTa model.

Adding Context Baruah et al. (2020) experiment with 5 forms of including context- not at all, including the last utterance from the dialogue, the last two utterances, the last three utterances, or the last three. They found that for Reddit data, using only the response performed the best, while for Twitter, having a context window that includes only the last utterance was the best-performing. We experiment with including the last post and context, and adding no context at all.

Following Baruah et al. (2020), the context is concatenated with the response in reverse order as they appear in the discourse, separated by a special token.

Ensemble Model We took inspiration from Lemmens et al. (2020) to design our ensemble system. In this architecture, two or more base models predict the label of a given instance, and the ensemble model—here, a decision tree—uses those predictions to inform its own output. Our D4 ensemble model was trained on the logit output (i.e., pre-softmax) of its base models on the SARC training set only. This model performed poorly on the SAND data, but further investigation indicated that it was not due to the use of logits, as opposed to probabilities. Even so, our final ensemble is trained on the probability output for a positive label (1), for *both* the SAND and SARC training sets.

4.1 Evaluation Methodology

We evaluate all of our systems using F1-score, as done in Khodak et al. (2018), as it considers both precision and recall. A true positive occurs when the correct class labels of "sarcastic" (1) and "non-sarcastic" (0) matches the predicted class label given by the model.

5 Approach

For both tasks, we first partitioned and reformatted the data so that it would be easily accessible during training. Then, we trained each system on a personal GPU and evaluated the systems on our development set using Patas.

5.1 Pre-processing

SARC's balanced training data is provided in .csv format. Each line includes the comment ID for the original post, IDs for any intervening comments, the two IDs of the sarcastic and non-sarcastic responses, and finally the label (0 or 1) for the respective responses. The full comment texts are available in a single JSON file, but because the full dataset comprises 533M comments, accessing this JSON file even once is time- and resource-intensive.

To combat this overhead, we loaded the full file once and printed smaller JSONs for the training and development sets, as described in Table 1. In the final JSONs, each line represents a single sarcastic or non-sarcastic response and contains the text for all previous comments and the response, as well as all the post IDs and the true label. An example is shown in Appendix B.

We use the native BERT/RoBERTa tokenizers, which prepend a [CLS] token, add a [SEP] token at the end of the text, and pad the input out to the maximum model length, 512. If the input is longer than the maximum model length, the remainder is truncated. We use a batch size of 8, so padding the input is necessary.

5.2 Model Training

Fine-tuning Each classifer model was trained on an NVIDIA RTX 2080 Ti GPU, and training lasted 4 hours. According to the suggestions of Devlin et al. (2019), we trained for 4 epochs. Training for many epochs could lead to catastrophic forgetting. To further mitigate this, we used the lowest suggested learning rate of 2e-5, as suggested in Sun et al. (2020). We keep the model checkpoint saved from the epoch with the best F1 score on the dev set.

Finetuning RoBERTa further on SAND took 2.5 hours per epoch on two NVIDIA RTX 2080 Ti GPUs. We only trained for 2 epochs, and kept the model from the first epoch as it had already begun to overfit.

Ensemble The ensemble models were trained on base model predictions, which required each base model to predict labels for the full training set(s). This step was run on a GPU node on Patas, the department's computing cluster. The iteration with longest prediction time was our SAND-finetuned RoBERTa, predicting on the SAND training data, which took about 14 hours. In general, we found that BERT models ran faster than RoBERTa, and iterations which used the SARC data as input ran faster than those using SAND.

Training the decision tree classifier took under a minute and did not require a GPU. Through trial and error, we found that shallower trees performed better, as they were less likely to overfit to the training data. Our D4 ensemble model has a maximum depth of 5. Our final ensemble model, trained on both tasks' training sets, has a maximum depth of 14.

5.3 Model Evaluation

We evaluated our models on the dev and test sets for either task. We created our primary task dev set using a randomly sampled 10% partition of the SARC balanced training data. We recorded F1-score for every system, as well as the predicted labels and gold standard labels to facilitate error analysis.

For D3, evaluating all 4 of our models on the SARC dev set took about 30 minutes on Patas. We

	Model	SARC Dev	SARC Test	SAND Dev	SAND Test
D2	RANDOM BASELINE	0.500	0.500	0.500	0.500
	BERT	0.725	–	–	–
D3	ROBERTA	0.729	–	–	–
	BERT + CONTEXT	0.727	–	–	–
	ROBERTA + CONTEXT	**0.736**	–	–	–
D4	SAND ROBERTA	0.0186	0.0236	**0.884**	**0.885**
	ENSEMBLE (BERT, ROBERTA)	0.728	**0.729**	0.116	0.118
Final	ENSEMBLE (2 ROBERTAS)	0.675	0.678	0.867	0.869

Table 3: F1 scores of each system on each evaluation subset. The best score for each subset is bolded. D2 and D3 systems were only evaluated on the SARC dev set. Only "SAND RoBERTa" and "Ensemble (2 RoBERTas)" were trained using both the SARC and SAND training sets. Otherwise, models were trained with SARC only.

requested and were allocated a GPU, though it is difficult to tell retroactively which node we were allocated and how much of our job time was spent on the GPU.

For D4, evaluating SAND RoBERTa on both dev and test splits of SARC and SAND took about 8 minutes each on two NVIDIA RTX 2080 Ti GPUs. Evaluation time for the ensemble model was dependent on its base models and the evaluation dataset. Evaluating with a SARC dataset, as opposed to SAND, and evaluating a BERT model, as opposed to RoBERTa, resulted in a quicker evaluation time. Running the ensemble itself is very quick, taking less than a minute. The total evaluation time for each base model ranged from 30 minutes to over 2 hours, so with two base models per ensemble, ensemble evaluation time ranges from 1 to 4 hours.

6 Results

The F1-scores for all iterations of our system are shown in Table 3 alongside a random baseline. The SARC-fine-tuned "RoBERTa + Context" from D3 performs best on the SARC dev set. We did not evaluate this model on the SARC test set, so while our D4 ensemble model performs best on that subset (0.729), we cannot conclude whether it outperforms the "RoBERTa + Context" model. Unsurprisingly, each of our D4 models appears to perform well on only one of the two tasks. "SAND RoBERTa" was most recently fine-tuned on the SAND training data for one epoch, and it performs best on the SAND task. The D4 ensemble was instead trained on the outputs of our SARC-fine-tuned models from D2 and D3, so it performed best on the SARC task.

Our final model uses the SARC-fine-tuned RoBERTa from D3 and the further SAND-fine-tuned RoBERTa from D4, and it generalizes best between the two tasks, with F1-scores above the random baseline for all evaluation sets.

7 Discussion

F1-scores for our primary task show improvement over the baseline as well as tentative improvement over model performance on the Reddit dataset in (Baruah et al., 2020). False positives for our baseline system are slightly more numerous than false negatives in the dataset, at 3769 and 3401, respectively. This breakdown can be seen in Figure 1.

Because the dataset is perfectly balanced, we would expect to see an even number of false positives and false negatives, suggesting a systematic error. One cause of this may be simple noise from the dataset: a lack of the /s tone indicator does not guarantee that an utterance is serious. For example, analysis of false positives found in the SARC dataset returns entirely un-serious phrases like the following:

> "It's obviously tracks from a giant water tractor, farming for giant arctic sea prawn!" [1]

> "Because OBAMMA IS A SERKIT MUSLIN." [75]

Table 3 displays a comparison of scores across model iterations. D3 showcases our first modifications to the initial BERT model with limited success. The best-performing model was "RoBERTA + Context," earning an F1 score of 0.736. The "BERT + Context" and no-context RoBERTa models individually performed at comparable levels in terms of F1 scores. However, further breakdown, as seen in Figure 1 shows that the no-context RoBERTa

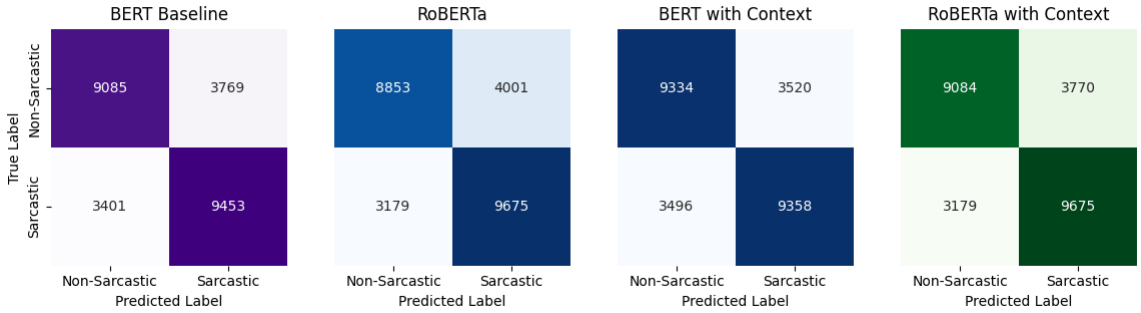

Figure 1: Heatmaps for our D2 initial system and our D3 models. These matrices represent evaluation only on the SARC dev set.

model may demonstrate a tendency to over-classify instances as sarcastic (n=13,676) compared to the BERT baseline (n=13,222) and BERT with context (n=12,878).

Of note, the combination of the two models resulted in identical numbers to the RoBERTA-only model for false negatives and true positives. It appears that the addition of context then "converted" some of the false positives into true negatives, resulting in an improved combined score. A similar phenomenon can be observed when comparing the Context-only model to the baseline; the addition of context led to a decrease in false positives (-249) and an increase in true negatives (+249) while other metrics were relatively consistent. It is unclear whether the addition of context in the D3 models contributes directly to an increase in true negatives, and if so, why context plays such an important role in identifying negative instances. One possible explanation is that context-agnostic models rely heavily on lexical cues that could signal sarcastic language in the absence of other context.

Our D4 iteration featured the introduction of the SAND dataset—for finetuning and evaluation—and the ensemble model. This iteration was characterized by a large disparity in performance between the two models. SAND RoBERTA performed the best on the SAND data it was trained on and also benefited from fine-tuning on the SARC data in previous iterations. The boost in F1 score, up to 0.885, may thus be an example of overfitting.

We reconciled the two models in a final ensemble model, which trained on a combination of the two datasets. Though there is a marginal drop in performance compared to training on a single dataset as in D4, the model had far greater success at switching between the two tasks. The confusion matrix in Figure 2 suggest the model was relatively balanced in positive (n=23,789) and negative (n=24,367) predictions on SAND data, but more often predicted negative (n=15,469) than positive (n=10,239) on SARC. Oddly, the reverse effect can be seen in D4, where the most successful model detected more negatives (n=27,154) than positives (n=21,032) for SAND.

The final ensemble continues to show a stark difference in performance between the two datasets, despite incorporating training data from both in the model. While the application of an existing model to a new task is expected to result in some disparities, the proximity of the adaptation task and primary task suggests that this difference in performance largely arose from a difference in data. One notable feature of the SAND data was its length; the average text length in the SAND dev set was 700 characters, compared to a mere 55 characters in the SARC dev set. We predict that this length was a significant determiner of model performance; since text input in excess of 512 characters is truncated by the model, a large number of SAND data instances were not processed completely. One possible explanation is that the end of the text posts is a crucial spot for markers of sarcasm. Another possible explanation is that the markers of sarcastic speech differ fundamentally between neurotypical and neurodivergent text; however, this has yet to be corroborated by further evidence.

Interestingly, all models tested on the four subsets in Table 3 performed slightly better on the test set compared to the dev set. However, the improvements are extremely minimal (+0.002), so there's little insight to gain from this observation.

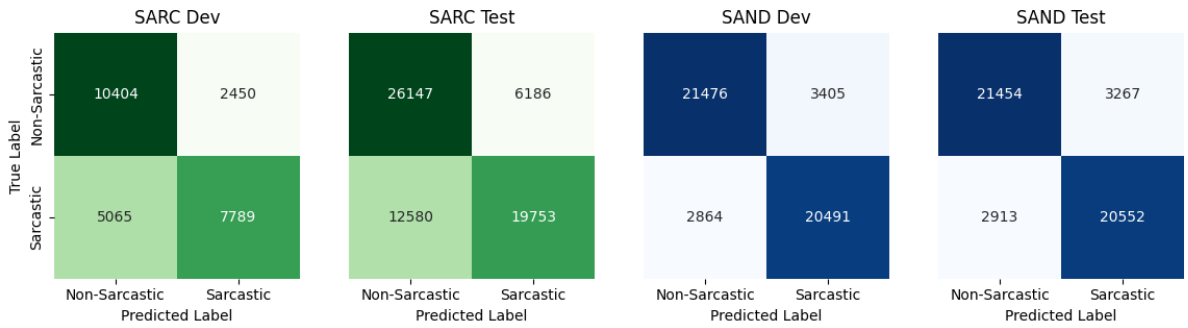

Figure 2: Heatmaps demonstrating the performance of our final ensemble model, which uses the SARC-finetuned and the SAND-finetuned RoBERTa models as base models. The two graphs on the left show the performance on the SARC task, while the two on the right show performance on our SAND corpus.

8 Ethical Considerations

Ethical concerns came into play at several points in the present study. Affiliation with online ND communities runs the risk of "outing" users as ND, something still heavily stigmatized in both private life and the workplace. Further, users featured in the SAND dataset are deprived of control over their own data, in the sense that they cannot edit or delete posts should their identity be exposed. As such, any concrete associations between posts and Reddit users' identities were carefully omitted. All usernames (including @ mentions) were removed from the SAND dataset prior to publishing on HuggingFace.

To our knowledge, this study presents the first exploration into sarcasm detection on primarily ND text. Thus, we must emphasize that no dataset can claim to be representative of neurodivergence as a whole, just as our dataset does not represent the true variation in ND communication styles, traits, opinions, and experiences. Our tasks and, by proxy, our systems are limited to English-only data, and though our system may generalize to text from other sources, it has only been trained on data from Reddit, whose users are themselves a subcommunity of the English-speaking and neurodivergent populations.

Further, the findings and dataset published here should not under any circumstances be used as a tool for diagnosis. Our approach seeks to compare the efficacy of conventional sarcasm detection systems on NT and ND text, but makes no attempt to classify that text into NT/ND categories.

9 Conclusion

Future work could build upon the present study by addressing sources of data noise mentioned throughout this paper: (1) sarcastic utterances that are not marked with tone indicators, and (2) overabundance of sarcastic utterances that are more confusing or ambiguous than usual. While collection of non-sarcastic text was limited to those users who had used tone indicators before, further selection for users who *consistently* used tone indicators, for example, a minimum of three times, may reduce the amount of incorrectly labeled instances.

Apart from data curation, future research could center on the correlation between text length and sarcasm detection performance. As one Reddit user noted in the SAND data, text length certainly has consequences for ND people:

> "Might want to bullet point what you need from us to get better responses, we do struggle with reading lengthy text." [post ID: czanycc]

This brings us to a final note: the task of sarcasm detection itself has important implications for online Neurodivergent communities. Though tone indicators are one method for expressing intent through online media, as we have seen in the present study, they are not a catchall solution. Sarcasm detection, difficult it may be, offers one more tool for disambiguating the complex world of online interactions. Neurodivergent people are but one community we must account for in order to build more inclusive and ultimately, more successful, systems.

References

Sheena K. Au-Yeung, Johanna K. Kaakinen, Simon P. Liversedge, and Valerie Benson. 2015. Processing of written irony in autism spectrum disorder: An eye-movement study. *Autism Research*, 8(6):749–760.

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, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Maria Febiana Christanti, Puri Bestari Mardani, and Khansa Ayu Fadhila. 2022. Analysing the meaning of tone indicators by neurodivergent community in twitter. *International Journal of Social Science Research and Review*, 5(1):5–15.

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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tamar Kalandadze, Courtenay Norbury, Terje Nærland, and Kari-Anne B Næss. 2018. Figurative language comprehension in individuals with autism spectrum disorder: A meta-analytic review. *Autism*, 22(2):99–117.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jens Lemmens, Ben Burtenshaw, Ehsan Lotfi, Ilia Markov, and Walter Daelemans. 2020. Sarcasm detection using an ensemble approach. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 264–269, Online. 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. *arXiv e-prints*, arXiv:1907.11692.

Amanda K Ludlow, Eleanor Chadwick, Alice Morey, Rebecca Edwards, and Roberto Gutierrez. 2017. An exploration of sarcasm detection in children with attention deficit hyperactivity disorder. *Journal of Communication Disorders*, 70:25–34.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification? *Preprint*, arXiv:1905.05583.

Piyoros Tungthamthiti, Kiyoaki Shirai, and Masnizah Mohd. 2016. Recognition of sarcasm in microblogging based on sentiment analysis and coherence identification. *Journal of Natural Language Processing*, 23(5):383–405.

Tiziana Zalla, Frederique Amsellem, Pauline Chaste, Francesca Ervas, Marion Leboyer, and Maud Champagne-Lavau. 2014. Individuals with autism spectrum disorders do not use social stereotypes in irony comprehension. *PloS one*, 9(4):e95568.

A Subreddits Scraped

Below is a list of the subreddits from which we collected our adaptation task data. We selected subreddits which we felt had primarily a neurodivergent userbase.

- r/ADHD
- r/adhdwomen
- r/aspergirls
- r/AutismTranslated
- r/autismmemes
- r/AutisticPride
- r/Autism
- r/AutisticAdults
- r/autisminwomen
- r/neurodivergent
- r/NeurodivergentLGBTQ

B Data Input Examples

Below are examples of SARC data after processing the data from .csv to JSON format. Post IDs are preserved for cross-referencing. Context is also preserved, as it is used as input in some of our experimental systems.

{"posts": ["Which one would you buy?"], "post_ids": ["yae7e"], "context_size": 1, "response_id": "c5u2m60", "response": "The $594.70+37.50$ version, because it obviously has more value.", "label": "1"}
{"posts": ["Which one would you buy?"], "post_ids": ["yae7e"], "context_size": 1, "response_id": "c5tsz3h", "response": "I'm surprised the second one doesn't have $640 shipping fees.", "label": "0"}

Below are examples of SAND data, stored in JSON format. We did not collect context for this dataset. Given more time, it would be possible to collect posts in context if our data collection method is reproduced, since parent posts are included in the original data. We organized our data as one dictionary, with post IDs as keys.

"fpij5l9": { "text": "I have this so badly. Its definitely my ADHD I started this thread partly hoping to feel less of a freak and wow its got at least 100 replies. Never had this many replies to a Reddit thread. You are so lucky not to have this one.", "label": 0}
"icgag9u": { "text": "If youre gay you get a day off of course. ", "label": 1 }

Our final system reads in each data format depending on the use of the --sand flag. Typically, our systems use the "posts" or "text" fields as input to the embedding layer, and "label" as the target output.