Utilizing Weak Supervision to Create S3D: A Sarcasm Annotated Dataset

Jordan Painter[1], Helen Treharne[1], and Diptesh Kanojia[1,2]
[1]Department of Computer Science, University of Surrey
[2]Surrey Institute for People-Centred AI, University of Surrey
United Kingdom
jp01166,h.treharne,d.kanojia@surrey.ac.uk

Abstract

Sarcasm is prevalent in all corners of social media, posing many challenges within Natural Language Processing (NLP), particularly for sentiment analysis. Sarcasm detection remains a largely unsolved problem in many NLP tasks due to its contradictory and typically derogatory nature as a figurative language construct. With recent strides in NLP, many pre-trained language models exist that have been trained on data from specific social media platforms, *i.e.,* Twitter. In this paper, we evaluate the efficacy of multiple sarcasm detection datasets using machine and deep learning models. We create two new datasets - a manually annotated gold standard Sarcasm Annotated Dataset (SAD) and a Silver-Standard Sarcasm-annotated Dataset (S3D). Using a combination of existing sarcasm datasets with SAD, we train a sarcasm detection model over a social-media domain pre-trained language model, BERTweet, which yields an F1-score of 78.29%. Using an Ensemble model with an underlying majority technique, we further label S3D to produce a weakly supervised dataset containing over $100,000$ tweets. We publicly release all the code, our manually annotated and weakly supervised datasets, and fine-tuned models for further research.

1 Introduction

Figurative language, such as the use of metaphors, irony and sarcasm, is ubiquitous in human communication, from ancient religious texts to social media micro texts. The detection of sarcasm in human communication is a challenging task where the goal is to identify sarcastic utterances from the data provided. There is no one definitive definition of sarcasm due to its nature as a language construct relying on factors such as domain and context, even regional differences (Dress et al., 2008), but a widely accepted definition is "a form of verbal irony that is intended to express contempt or ridicule" (Joshi et al., 2017).

Sarcasm has a diminishing effect on sentiment analysis due to sarcastic text often having the opposite implied meaning to a literal word-for-word meaning of the text (Pang and Lee, 2008). For example, *"I just love it when my flight gets delayed for 4 hours"*, is clearly sarcastic, as using the word "love" to express feelings on something rather inconvenient would be unusual outside of a sarcastic context. Such challenges demonstrate the importance of recognising sarcasm in social media (Farhadloo and Rolland, 2016), as recognising the potential for a given text utterance to be sarcastic can bridge the gap in human-machine communication. The NLP research community has investigated the detection of sarcasm using various machine/deep learning approaches (Potamias et al., 2019; Ghosh and Veale, 2016; Reyes and Rosso, 2011; Wankhade et al., 2022). Several datasets exist for the task of sarcasm detection using text (Riloff et al., 2013; Ptácek et al., 2014; Van Hee et al., 2018; Khodak et al., 2017) as well as multimodal datasets (Castro et al., 2019; Ray et al., 2022), which support the extraction of features from video and speech. Transformer (Vaswani et al., 2017) based language models have shown to perform very well for classification tasks in various NLP sub-areas, and a number of BERT (Devlin et al., 2018a) based language models have been released which can help perform this NLP task.

In this paper, we attempt to collate these efforts for the task of sarcasm detection. We restrict our focus to the detection of sarcasm on a social media platform, *i.e.,* Twitter. Initially, we curated our dataset (SAD) by crawling for tweets and labelling them with the help of two annotators. We extensively evaluate machine and deep learning-based approaches on various existing datasets and our dataset. We apply standard pre-processing and combine all the datasets to evaluate several classification approaches. Using an Ensemble of the best language models trained over the largest datasets, we further label 100K tweets to create Silver-Standard Sarcasm-annotated Dataset (S3D). The key contributions of our work are as follows: 1) A sarcasm-annotated dataset (SAD) of social

Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS), pages 197 - 206

media microblogs, 2) Performance evaluation of various existing language models for the binary classification task of sarcasm detection, 3) Curation and weak-supervision-based labelling for a silver-standard sarcasm-annotated dataset (S3D), 4) Release of code, data, and models created on Github, and HuggingFace platforms, publicly, for the research community[1].

This paper is organised as follows. Section 2 briefly describes previous approaches to sarcasm detection. Section 3 describes our chosen datasets and their sources. Section 4 explains the methodology behind the proposed experiments, summarising the approaches for our machine learning and deep learning experiments. Section 5 discusses choices made for running our experiments, Section 6 discusses the results of these experiments in detail, along with the approach used to obtain a new weakly supervised dataset.

2 Related Work

Transformer-based approaches have increased in prevalence within NLP and also within sarcasm detection literature. This is most notably due to their ability to accurately pick up semantic and syntactic relationships within a piece of text. Joshi et al. (2017) discuss various approaches to the task of sarcasm detection including rule-based and machine learning-based, and also discusses sarcasm from the linguistics perspective. Shangipour ataei et al. (2020) dicusses several approaches to perform sarcasm detection. These include a BERT (Devlin et al., 2018b) model with no concatenated layers, BERT encodings with a Logisitic Regression model, and other language models such as IAN (Ma et al., 2017) which are trained and evaluated on a Twitter-based sarcasm dataset. In these experiments, the BERT language model with no added layers performs the best on the dataset, achieving an F1-score of 73.4. Some existing literature investigates methods for performing sarcasm detection in Arabic (Abu Farha and Magdy, 2021), where a multitude of Transformers are used, including mBERT, XLM-RoBERTa (Conneau et al., 2020) and language-specific models like MARBERT (Abdul-Mageed et al., 2021). The best model in this research achieves an F1-score of 58.4 in a low-resource scenario. In Potamias et al. (2019), an RCNN-RoBERTa methodology was proposed, where a RoBERTa transformer was utilized with BiLSTM to improve upon F1-scores from state-of-the-art neural network classifiers on the dataset released with the SemEval 2018 Shared Task 3 (Van Hee et al., 2018). This paper also reports that the RCVV-RoBERTa approach achieved an F1-score of 90.0 on the Riloff dataset (Riloff et al., 2013). Ghosh and Veale (2016) demonstrate a variety of results on a Twitter dataset, training a collection of architectures involving Convolution Neural Network (CNN) and Long-Short Term Memory (LSTM) to achieve an impressive F1-score of 92.1 with their best configuration. An Ensemble approach was demonstrated in Goel et al. (2022) where a weighted average Ensemble of a CNN, an LSTM and a Gated Recurrent Unit (GRU) based architectures are trained with GloVe (Pennington et al., 2014) word embeddings in order to identify sarcasm, showing that the Ensemble outperformed others by up to 8% on SARC (Khodak et al., 2017), a Reddit comments dataset.

Machine learning approaches have decreased in popularity due to the improvements shown by Transformers-based architectures in recent developments. Earlier approaches to sarcasm detection include Reyes and Rosso (2011) and Barbieri et al. (2014) that used a Naive Bayes and Decision Tree model, respectively, in order to identify sarcasm where both achieve the best F1-scores over 70 on their chosen datasets.

To curate sarcasm-annotated datasets, one can perform manual annotation, which involves a significant cost in terms of time and money. Moreover, manual annotations for subjective linguistic constructs like sarcasm are questionable unless multiple annotators label the data, and an almost perfect inter-annotator agreement can be seen within the labelling. An example of this approach is the creation of the Riloff dataset (Riloff et al., 2013). On the other hand, sarcasm research has also utilised 'self-annotated tags' from social media forums, such as '#sarcasm' from tweets and '/s' in Reddit comments. Such data collection methods can be automated, and a large amount of data can easily be collected. However, the quality of such datasets in terms of label accuracy can be questioned. Self-annotation was used in the creation of the Ptacek dataset (Ptácek et al., 2014) from English tweets, and the creation of the SARC dataset (Khodak et al., 2017) from Reddit comments. However, we follow a hybrid approach as we collect SAD using '#sarcasm' from Twitter and then manually label it.

[1] https://github.com/surrey-nlp/S3D

A limitation of publicly available datasets based on tweet IDs, *e.g.,* Riloff et al. (2013) is that the tweet data retrieval based on the IDs can diminish over time. If a significant number of tweets are deleted, then it would not be possible to reproduce the results on the original dataset. In *e.g.,* Riloff et al. (2013), the number of tweets, at the time of writing the paper, that can be retrieved related to the IDs in the dataset is 710 compared to the original 3000 data instances. The contribution of our weak supervision-based approach is to help produce labelled data, the benefit of which could be to augment existing datasets that have diminished over time with automatically labelled data or also to create new silver standard datasets.

3 Datasets

We test our proposed approach for sarcasm detection on a total of six datasets, summarised in Table 1. Four of these data sets are benchmark datasets retrieved from either Twitter or Reddit summarised below: **SARC:** The only benchmark Reddit dataset we use is the SARC dataset (Khodak et al., 2017), a vast corpus of self annotated comments that were collected taking advantage of the '/s' tag that Reddit users can insert at the end of a comment to denote sarcasm. **Ptacek:** In Ptácek et al. (2014) an English and Czech sarcasm dataset was released to demonstrate the applicability a machine learning approach for sarcasm detection. For our proposed experiments the English dataset was used, which was curated collecting self-annotated tweets containing the #sarcasm hashtag. **SemEval2018:** We use the SemEval 2018 Task 3 dataset, which is a manually annotated Twitter dataset that was released for the SemEval 2018 Irony Detection in English Tweets shared task (Van Hee et al., 2018). **Riloff:** We use the dataset released by Riloff et al. (2013), which was manually annotated for sarcasm in order to train a bootstrapping algorithm on positive sentiment phrases and negative situation phrases from sarcastic tweets.

3.1 Our Dataset (SAD)

The first new dataset we introduce is the SAD dataset, a collection of scraped tweets containing a total of 2,340 data points, 1,170 of which are initially self-annotated for sarcasm through selecting tweets that contained the #sarcasm hashtag.

The TWINT[2] library was used to search for tweets that contained a #sarcasm hashtag, which was stored along with other relevant data points, including the respective tweet ID and username associated with the said tweet. Within the dataset, we ensured that there was one sarcastic and one non-sarcastic tweet for each unique username. We used TWINT to scrape and identify a second tweet for each user name to achieve this.

This resulted in several tweets, which were manually labelled by two annotators to ensure label accuracy and the presence of sarcasm; while ensuring that the tweet is not just a list of hashtags attached to a link to an image or website - a common spamming method on Twitter. To assign the final class label on disputed data instances, we requested a third annotator to go through the tweet and assign a class label (without looking at any of the previous annotations). We obtain an inter-annotator agreement score of 0.83 (Cohens' Kappa) where the *p-value* was < 0.05 which signifies almost perfect agreement. We also compared the manually labelled sarcastic tweets with the self-annotations in the same tweets, and 98% matches were observed.

3.2 Combined Dataset

The second dataset is a new 'Combined' dataset. This collates the four benchmark datasets and the new SAD dataset. This resulted in a corpus of 1,022,546 entries of labelled text, both taken from Reddit and Twitter, where an approximate split of 50/50 sarcastic to non-sarcastic text was achieved. We hypothesise that *various domains of sarcastic text present in multiple datasets should help a computational model generalise better and learn to identify sarcastic instances*. We perform similar experiments on this dataset to generate sarcasm detection models and evaluate over its test set.

3.3 Dataset Statistics and Validation

In Table 1, there is a clear difference between the size of each of the datasets. Most noticeably, the SARC dataset has over 1,000,000 entries, in comparison to the Riloff dataset, which has less than 1,000. Most of the datasets are balanced to an approximate 50% split for sarcastic and non-sarcastic text alike.

In the case of the Riloff and Ptacek datasets, both available versions online only contained the tweet IDs and their respective labels, meaning they were

[2]TWINT website: https://github.com/twintproject/twint

Dataset	Total	Training	Validation	Testing	Sarcastic	Non-Sarcastic
SARC	1,010,773	707,541	151,616	151,616	505,368	505,405
Ptacek	4,906	3,434	736	736	2,781	2,125
SemEval	3,817	2,671	573	573	1,901	1,916
Riloff	710	497	106	107	160	550
SAD (Our Dataset)	2,340	1,638	351	351	1,170	1,170
Combined	1,022,546	715,782	153,382	153,382	511,380	511,166

Table 1: Table demonstrating the Train/Valid/Test and Sarcastic/Non-sarcastic splits of the chosen datasets

collected by using Tweepy, the Python library used for accessing Twitter's API. This, unfortunately, meant that out of the 3,000 tweets available in the original Riloff dataset, only 710 were able to be retrieved, as when a user deletes their account or a specific tweet, it can no longer be retrieved.

3.4 Preprocessing

For the pre-processing of the chosen datasets, all were first checked through to delete null values that were in place of comments. This was followed by all text being transformed to lowercase. Every data entry was then checked for the presence of a #sarcasm hashtag, which we would then remove. Datasets such as the Ptacek and SAD datasets that use self-annotation to find sarcastic tweets would have this hashtag in every sarcastic entry. Therefore, they needed to be removed to ensure none of our models would make predictions based on the presence of this hashtag alone. Every username present in the Twitter datasets was replaced with '@user' to reduce unnecessary noise from a large number of unique usernames. As a final measure, all URLs and remaining punctuation were also removed from each comment to reduce noise further.

3.5 Evaluation Metrics

The primary evaluation metric of the proposed experiments is the F1-score of the sarcastic. This metric is necessary over binary accuracy due to the typical imbalanced nature of sarcasm detection datasets. Both the precision and recall scores of the sarcastic class are also recorded within Section 6.

4 Methodology

For our machine learning experiments we use DT (Laurent and Rivest, 1976) and LR (Cox, 1958) models. Our approaches to vectorising text for feature extraction utilise Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Word2Vec is a model architecture for computing vector representations of words from text, as is GloVe, which has an additional focus on Latent Semantic Analysis.

For our deep learning based experiments, a total of five pre-trained language models were used: BERT (Devlin et al., 2018a), RoBERTa$_{base}$ & RoBERTa$_{large}$ (Liu et al., 2019), Twitter-RoBERTa (Barbieri et al., 2020) and BERTweet (Nguyen et al., 2020).

BERT was introduced as a state-of-the-art transformer that improved results on multiple benchmarked NLP tasks. The language model was demonstrated as being able to be fine-tuned to create models for a wide range of tasks including question inference and next sentence prediction. **RoBERTa** was built on BERT through modifying key hyper-parameters and removing the next-sentence-prediction pre-training objective, on top of training with much larger batches and learning rates. The RoBERTa$_{large}$ configuration follows the same architecture but contains more hidden units and twice the number of encoder layers. **Twitter-RoBERTa** was introduced as RoB-RT by Barbieri et al. (2020) and is a RoBERTa$_{base}$ model that was trained on a total of 60M tweets, consisting of 584 million individual tokens. **BERTweet** has the same architecture of BERT-base and is trained on an 80GB corpus of 850M English tweets.

Each of these models was fine-tuned for the purpose of sarcasm detection. The fine-tuning process comprises adding a dropout layer on top of the pre-trained model, followed by a fully connected layer which was then fed into a final layer using a *softmax* activation function for classification.

5 Experiment Setup

As discussed in Section 4, the experiments have been split into the two categories of machine

	Word2Vec+LR			Word2Vec+DT			GloVe+LR			GloVe+DT		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
SARC	62.93	61.06	**61.98**	57.59	55.57	56.56	62.06	56.56	58.63	56.69	55.34	56.02
Ptacek	72.31	71.80	72.06	64.43	61.37	62.86	75.96	74.88	**75.41**	66.58	62.32	64.38
SemEval	63.57	59.79	**61.62**	53.71	53.14	53.43	60.47	54.54	57.35	53.28	53.84	53.56
Riloff	100	03.57	06.89	17.39	14.28	15.68	85.71	21.42	34.28	39.13	32.14	**35.29**
SAD	62.14	55.56	58.67	63.38	58.58	**60.89**	60.87	56.57	58.64	65.48	55.56	60.11
Combined	62.15	55.56	58.67	56.96	55.05	56.56	61.69	60.25	**60.96**	56.33	55.25	55.78

Table 2: Results of Sarcasm Detection experiments with Machine Learning approaches, where *P* denotes Precision, *R* denotes Recall and *F1* denotes the F1-score of the experiment. Underlined results denote the best F1-score for each model. Results in bold denote the best F1-score for its own dataset

	BERT			BERTweet			RoBERTa$_{base}$			Twitter-RoBERTa			RoBERTa$_{large}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
SARC	73.91	79.47	76.59	76.52	80.35	**78.39**	76.23	78.35	77.30	74.89	80.52	77.61	77.65	77.57	77.61
Ptacek	84.46	75.83	79.99	88.86	85.07	86.92	88.41	88.63	88.52	91.46	86.26	88.78	91.50	89.33	**90.41**
SemEval	59.61	74.83	66.36	69.81	77.62	73.51	78.42	90.21	83.90	78.37	87.41	82.64	81.11	87.06	**83.98**
Riloff	66.67	35.71	46.51	85.71	42.86	**57.14**	58.33	50.00	53.85	55.56	53.57	54.54	85.71	42.86	**57.14**
SAD	65.89	71.21	68.45	77.36	62.12	68.91	81.49	93.43	**87.06**	82.19	90.90	86.33	86.84	83.33	85.05
Combined	76.46	75.36	75.91	75.99	80.72	**78.29**	76.00	78.48	77.22	76.68	77.72	77.19	76.15	79.95	78.01

Table 3: Results of Sarcasm Detection experiments with Deep Learning approaches, where *P* denotes Precision, *R* denotes Recall and *F1* denotes the F1-score of the experiment. Underlined results denote the best F1-score for each model. Results in bold denote the best F1-score for its own dataset

learning-based and deep learning-based experiments. The environment used to run the machine learning experiments was a Kaggle notebook, whereas the deep learning experiments were run on an i9 machine with 2 NVIDIA RTX A5000 GPUs.

5.1 Hyper-parameter Setting

For the machine learning experiments, both the DT and LR models were trained with the default hyperparameters as set in the scikit-learn[3] (Pedregosa et al., 2011) library. For the deep learning experiments, every configuration had the same set of hyper-parameters apart from one exception in the batch size. The batch size was set to 32 for all of the language models except for RoBERTa$_{large}$, where the batch size was set to 4. This was due to the computational limitations that arose due to RoBERTa$_{large}$ being trained on the exceptionally large SARC and 'Combined' datasets with a batch size of 32. Every configuration had a learning rate of 3e-6, with an Adam activation function. The output of each language model was fed into a dropout layer of 0.3, and followed by a hidden layer with a ReLU activation function and 256 hidden units. Finally, the output of the hidden layer was fed through a Softmax activation function with 2 units to perform binary classification.

6 Results and Discussion

Table 2 and 3 show the results of the machine learning and deep learning-based experiments, respectively. According to Table 2, it is clear that the success of each respective machine learning approach is highly dependent upon the particular dataset on which it is being trained. The Ptacek dataset has the highest F1-scores for sarcasm detection for each machine learning approach, as can be seen by the underlined results, and also achieves the highest F1-score in the entire set of experiments (75.41) when used with the GloVE+LR model.

Table 4 demonstrates that for the Word2Vec+DT (worst) and GloVe+LR (best) models, there is no consistency in how negative phrases such as "didn't think", "didn't realise" are labelled compared to the actual label used within the dataset. The last extract was labelled incorrectly by both models, with neither understanding that the word "love" was being used in a sarcastic context, which could be seen as a limitation of the machine learning approaches. Although, without context, it is fair to assume that the user could have been non-sarcastic in this tweet.

[3] https://scikit-learn.org

Comment	Word2Vec+ DT Label	GloVe+ LR Label	Ground Truth
'didnt realize @user referees were so fluent in russian'	1	1	1
'well hello depression nice to see ya again didnt think youd stay away' much longer	0	1	1
'dont you just love the hip hop music and club music they played in the background of the @user movie i do'	0	0	1

Table 4: Entries from the Ptacek dataset labelled by the highest and lowest scoring ML experiments and their ground truth labels. 1 represents a sarcastic label and 0 represents a non-sarcastic label.

The SemEval dataset achieves its highest F1-score of 61.62 using the Word2Vec+LR model. The Riloff dataset has the weakest set of F1-scores across each approach, with it is best F1-score (35.29) still being lower than any F1-score for any other dataset. Interestingly, the Word2Vec+LR model achieves a perfect precision score, whereas the associated scores for this model are the lowest for all experiments.

From Table 2, it is seen that our SAD dataset achieves similar F1-scores across each model, with a variance of 2.25 between the highest and lowest scores. The SAD dataset and the Riloff dataset are the only two out of the six to achieve their best scores from a decision tree classifier as opposed to a logistic regression classifier.

From Table 3, we observe the best F1-score for the task of sarcasm detection using deep learning methods is 90.41 on the Ptacek dataset with the use of the RoBERTa$_{large}$ language model. As is seen with our machine learning approaches, Ptacek again is the dataset for which all of our models achieve the highest F1-scores. The Ptacek dataset has only 736 test set instances and may not have particularly challenging sarcasm examples. We make this assumption based on the performance of the same pre-trained language models on much larger datasets, viz., SARC (78.39) and Combined (78.29). The RoBERTa$_{large}$ language model achieves the highest F1-score of 83.98 on the SemEval dataset.

There is more success with the unbalanced Riloff dataset within the deep learning experiments as opposed to the machine learning experiments. The lowest F1-score using the Riloff dataset in Table 3 (46.51) achieved by our BERT model is still higher than the highest F1-score in Table 2 (35.29) from the GloVe+DT model. The results achieved are again lower than the results obtained from the rest of our chosen datasets. Both the BERTweet and RoBERTa$_{large}$ language models incidentally achieve the exact same precision, recall and F1-scores (57.14) on this dataset.

Our SAD dataset has high F1-scores across each model, 87.06 being the highest achieved by the RoBERTa$_{base}$ language model. The BERT language model achieves the weakest F1-score on the dataset (68.45), followed closely by the BERTweet model (68.91). This was unexpected as the BERTweet language model was pre-trained only on tweets. Further unexpectedly, the RoBERTa$_{base}$ model actually achieves the best overall F1-score on the SAD dataset, despite the model not being pre-trained on any tweets at all. This performance may be attributed to the significantly larger dataset used for training the RoBERTa model.

Ironically, despite being pre-trained solely on $850M$ tweets, the BERTweet model achieves the highest F1-score of 78.39 on the SARC dataset, the only dataset that does not include any tweets.

From Table 3, we also observe that the BERTweet and RoBERTa$_{large}$ language models outperform every other approach. They achieve the highest F1-score on three datasets, respectively. For the SARC and the 'Combined' dataset, the BERTweet analysis provides the best F1-scores, and these datasets are, in fact, the largest datasets. Furthermore, the BERTweet language model has the advantage of being pre-trained specifically on data consisting of tweets, as opposed to the less focused domain data that was used to train RoBERTa$_{large}$. We hypothesise that the fine-tuned sarcasm detection models trained over large datasets would be able to generalise better as the training sets would also be large.

Comment	BERTweet Label	Ground Truth
'more fragmentation is exactly what we need in mobile payments'	1	1
'hockey wouldnt work in quebec city'	1	1
'this is new and interesting'	1	1
'i call them suckers'	1	0
'by doing the same thing i do every night and day nothing'	0	1
'huge moves were making gonna take this league by storm'	0	1

Table 5: Entries from the 'Combined' dataset with their predicted labels by our pre-trained BERTweet model and their ground truth labels. 1 represents a sarcastic label, and 0 represents a non-sarcastic label.

Table 5 shows the labels predicted by the model

trained using the BERTweet model on the 'Combined' dataset. The first three entries show examples of correctly identified sarcasm. If taken literally, the third entry could be considered as a genuine statement, but the model determines this to be sarcastic, and in fact, it is labelled as such within the dataset.

There are entries where the model incorrectly labels sarcastic text extracts. In the fourth row, an instance of a false positive can be seen, where our pre-trained model incorrectly determines a tweet is sarcastic when it was not labelled as such. The word "suckers" might indicate some humorous intent to the text, implying sarcasm may be used in the comment.

The last two entries in Table 5 are examples of labelled sarcasm that our model did not determine to be sarcastic. The fifth entry puts forward an unlikely proposition similar to the first two entries in that it is probably untrue that the user spends all night and day doing nothing.

Although the model made the correct prediction in the rather specific domain of "quebec" and "hockey", it makes an incorrect prediction in this broader context. This is demonstrable of how figurative language and the understanding of such truly rely on contextual differences. These contextual differences impact human, and, particularly, machine understanding of sarcasm. Again, this struggle of the models' prediction capabilities in a broader context is seen in the final entry, where the user has intended the text to be sarcastic, but it has not been labelled by our BERTweet model as such. Even with this small scope of examples where our model has made incorrect predictions, our fine-tuned BERTweet model is still our highest-scoring language model on our largest datasets, and thus we will use fine-tuned BERTweet models for the purpose of labelling a weakly supervised dataset.

6.1 S3D Dataset: Using Weak Supervision

The results for the analysis of the fine-tuned BERTweet model for both the SARC and 'Combined' datasets are very similar, but we note that the 'Combined' dataset contains both Tweets and Reddit comments. Similarly, RoBERTa$_{large}$ model performs well on the Combined dataset (78.01). We create *an Ensemble model using the majority voting technique and utilise these three variants* - a BERTweet model trained on SARC and Combined datasets, and a RoBERTa$_{large}$ model trained on the combined dataset. We further use this Ensemble model to label our new dataset, the curation for which is described below.

We used the TWINT package to scrape a total of $100,000$ tweets[4] to be labelled by our chosen model. We call this a silver-standard sarcasm annotated dataset 'S3D'. Every tweet was pre-processed as described in section 3.4, then encoded using the BERTweet model. Our Ensemble model was then used to generate predictions on the pre-processed $100,000$ tweets. The results of this labelling process are shown in Table 6.

Sarcastic	Non-Sarcastic	Total
38879	61121	100000

Table 6: Number of sarcastic and non-sarcastic labels generated by our pre-trained BERTweet model

Out of $100,000$ tweets chosen at random, nearly 40% were considered by our model to contain sarcasm. We show excerpts from this dataset in Table 7.

Comment	Label
'@user you look soo freaking good in the poster man'	1
'tweet of the year @user you make sense'	1
'i bet theres no dry eyes leaving the concert' tonight	1
'the best joke yet'	1
'wow the war just ended i didnt know that'	1
'truly changed the trajectory of my life'	1
'yes a lot of great things will happen in the next 3 months'	1

Table 7: Entries from the S3D dataset, each labelled as sarcastic by our fine-tuned BERTweet language model. 1 represents a sarcastic label and 0 represents a non-sarcastic label.

Several entries seen in Table 7 could equally be seen as extracts with genuine sentiment as much as they could be sarcastic. The first entry is an example of this as if taking the tweet at its face value without context, it is very possible the user is being honest and complementing another user on the platform. Take the sixth entry, which could again be just as authentic as it could be sarcastic. To decide for ourselves, we would need to view some context as to what the event is that the user

[4]This set of collected tweets were posted between 7 September 2022 and 9 September 2022

is referring to. If the subject matter was serious, it is fair to assume the user is not being sarcastic. Some excerpts such as the second entry are perhaps more obviously sarcastic, as reminding someone they make sense while also awarding them "tweet of the year" carries a more disingenuous sentiment. The same could be said for the fifth entry, where it is very unlikely the user is being genuine about being unaware of the topic mentioned in tweet.

We also performed a simple exploratory experiment where we concatenate S3D with the 'Combined' dataset and perform fine-tuning with the help of the BERTweet model. A simple fine-tuning experiment with the same hyperparameters *achieves the best F1-score of* 78.87*, which is an improvement on the scores reported earlier on both SARC and 'Combined' datasets.* The reported precision and recall scores were 78.84 and 78.89 respectively. This shows the efficacy of our weakly supervised S3D dataset.

7 Conclusion and Future Work

In this paper, we utilise several existing machine- and deep learning-based approaches to perform the task of sarcasm detection over various datasets. From a social media platform, we curate and manually label a sarcasm dataset and benchmark its efficacy with these approaches. We also perform an exhaustive evaluation with the help of pre-trained language models, including some models specifically trained using social media data. Using an Ensemble model based on multiple fine-tuned BERTweet models, we labelled an additional $100,000$ tweets and release this silver-standard sarcasm annotated corpus, called S3D. We also perform a fine-tuning experiment after concatenating S3D with the 'Combined' dataset and achieve the best F1-score of 78.87 over the large datasets discussed in this paper. By contributing a weak supervision-based approach, we facilitate the automatic production of labelled data that can be used to augment existing datasets or create new silver standard datasets. We also release the code, the manually labelled dataset, and models created with our experiments publicly for further research.

In future, we would like to perform a more fine-grained annotation for sarcasm with sub-categories as defined in existing linguistic literature. We also aim to perform similar experiments for multimodal sarcasm detection in order to contribute further resources to the community.

Limitations and Biases

Our work releases two datasets for modelling sarcasm from social media posts but they may contain biases as present in any raw social media dataset.

Ethics Statement

We ensured that while curating our SAD and S3D datasets, information relating to the originator of the tweet was removed, and all user-specific information contained within a tweet, for example, usernames and user IDs, was removed during preprocessing to preserve anonymity. Similarly, information regarding the time of posting and location was removed during curation. The released datasets only contain tweet IDs along with their respective sarcasm labels, again to ensure the anonymity of our datasets.

References

Muhammad Abdul-Mageed, AbdelRahim A. Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: deep bidirectional transformers for arabic. *CoRR*, abs/2101.01785.

Ibrahim Abu Farha and Walid Magdy. 2021. Benchmarking transformer-based language models for Arabic sentiment and sarcasm detection. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 21–31, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Francesco Barbieri, José Camacho-Collados, Leonardo Neves, and Luis Espinosa Anke. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *CoRR*, abs/2010.12421.

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014. Modelling sarcasm in Twitter, a novel approach. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–58, Baltimore, Maryland. Association for Computational Linguistics.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an _Obviously_ perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4619–4629, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised

cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

David R Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Megan L. Dress, Roger J. Kreuz, Kristen E. Link, and Gina M. Caucci. 2008. Regional variation in the use of sarcasm. *Journal of Language and Social Psychology*, 27:71 – 85.

Mohsen Farhadloo and Erik Rolland. 2016. Fundamentals of sentiment analysis and its applications. In *Sentiment analysis and ontology engineering*, pages 1–24. Springer.

Aniruddha Ghosh and 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, San Diego, California. Association for Computational Linguistics.

Priya Goel, Rachna Jain, Anand Nayyar, Shruti Singhal, and Muskan Srivastava. 2022. Sarcasm detection using deep learning and ensemble learning. *Multimedia Tools and Applications*.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: A survey. *ACM Comput. Surv.*, 50(5).

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A large self-annotated corpus for sarcasm.

Hyafil Laurent and Ronald L Rivest. 1976. Constructing optimal binary decision trees is np-complete. *Information processing letters*, 5(1):15–17.

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. *CoRR*, abs/1907.11692.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4068–4074.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

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, Online. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1–2):1–135.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher 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, Doha, Qatar. Association for Computational Linguistics.

Rolandos Alexandros Potamias, Georgios Siolas, and Andreas-Georgios Stafylopatis. 2019. A transformer-based approach to irony and sarcasm detection. *CoRR*, abs/1911.10401.

Tomás Ptácek, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on czech and english twitter. In *COLING*.

Anupama Ray, Shubham Mishra, Apoorva Nunna, and Pushpak Bhattacharyya. 2022. A multimodal corpus for emotion recognition in sarcasm. In *Proceedings of the Language Resources and Evaluation Conference*, pages 6992–7003, Marseille, France. European Language Resources Association.

Antonio Reyes and Paolo Rosso. 2011. Mining subjective knowledge from customer reviews: A specific case of irony detection. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 118–124, Portland, Oregon. Association for Computational Linguistics.

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, Seattle, Washington, USA. Association for Computational Linguistics.

Taha Shangipour ataei, Soroush Javdan, and Behrouz Minaei-Bidgoli. 2020. Applying transformers and

aspect-based sentiment analysis approaches on sarcasm detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 67–71, Online. Association for Computational Linguistics.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7):5731–5780.