A Report on the 2020 Sarcasm Detection Shared Task

Debanjan Ghosh[1], Avijit Vajpayee[1], and Smaranda Muresan[2]
[1] Educational Testing Service
[2]Data Science Institute, Columbia University
{dghosh, avajpayee}@ets.org
smara@columbia.edu

Abstract

Detecting sarcasm and verbal irony is critical for understanding people's actual sentiments and beliefs. Thus, the field of sarcasm analysis has become a popular research problem in natural language processing. As the community working on computational approaches for sarcasm detection is growing, it is imperative to conduct benchmarking studies to analyze the current state-of-the-art, facilitating progress in this area. We report on the shared task on sarcasm detection we conducted as a part of the 2nd Workshop on Figurative Language Processing (FigLang 2020) at ACL 2020.

1 Introduction

Sarcasm and verbal irony are a type of figurative language where the speakers usually mean the opposite of what they say. Recognizing whether a speaker is ironic or sarcastic is essential to downstream applications for correctly understanding speakers' intended sentiments and beliefs. Consequently, in the last decade, the problem of irony and sarcasm detection has attracted a considerable interest from computational linguistics researchers. The task has been usually framed as a binary classification task (sarcastic vs. non-sarcastic) using either the utterance in isolation or adding contextual information such as conversation context, author context, visual context, or cognitive features (Davidov et al., 2010; Tsur et al., 2010; González-Ibáñez et al., 2011; Riloff et al., 2013; Maynard and Greenwood, 2014; Wallace et al., 2014; Ghosh et al., 2015; Joshi et al., 2015; Muresan et al., 2016; Amir et al., 2016; Mishra et al., 2016; Ghosh and Veale, 2017; Felbo et al., 2017; Ghosh et al., 2017; Hazarika et al., 2018; Tay et al., 2018; Oprea and Magdy, 2019; Majumder et al., 2019; Castro et al., 2019; Ghosh et al., 2019).

In this paper, we report on the shared task on sarcasm detection that we conducted as part of the

Turns	Message
$Context_1$	The [govt] just confiscated a \$180 million boat shipment of cocaine from drug traffickers.
$Context_2$	People think 5 tonnes is not a lot of cocaine.
$Response$	Man, I've seen more than that on a Friday night!

Table 1: Sarcastic replies to conversation context in Reddit. $Response$ turn is a reply to $Context_2$ turn that is a reply to $Context_1$ turn

2nd Workshop on Figurative Language Processing (FigLang 2020) at ACL 2020. The task aims to study the role of conversation context for sarcasm detection. Two types of social media content are used as training data for the two tracks - microblogging platform such as Twitter and online discussion forum such as Reddit.

Table 1 and Table 2 show examples of three turn dialogues, where $Response$ is the sarcastic reply. Without using the conversation context $Context_i$, it is difficult to identify the sarcastic intent expressed in $Response$. The shared task is designed to benchmark the usefulness of modeling the entire conversation context (i.e., all the prior dialogue turns) for sarcasm detection.

Section 2 discusses the current state of research on sarcasm detection with a focus on the role of context. Section 3 provides a description of the shared task, datasets, and metrics. Section 4 contains brief summaries of each of the participating systems whereas Section 5 reports a comparative evaluation of the systems and our observations about trends in designs and performance of the systems that participated in the shared task.

Turns	Message
$Context_1$	This is the greatest video in the history of college football.
$Context_2$	Hes gonna have a short career if he keeps smoking . Not good for your health
$Response$	Awesome !!! Everybody does it. That's the greatest reason to do something.

Table 2: Sarcastic replies to conversation context in Twitter. $Response$ turn is a reply to $Context_2$ turn that is a reply to $Context_1$ turn

2 Related Work

A considerable amount of work on sarcasm detection has considered the utterance in isolation when predicting the sarcastic or non-sarcastic label. Initial approaches used feature-based machine learning models that rely on different types of features from lexical (e.g., sarcasm markers, word embeddings) to pragmatic such as emoticons or learned patterns of contrast between positive sentiment and negative situations (Davidov et al., 2010; Veale and Hao, 2010; González-Ibáñez et al., 2011; Liebrecht et al., 2013; Riloff et al., 2013; Maynard and Greenwood, 2014; Joshi et al., 2015; Ghosh et al., 2015; Ghosh and Muresan, 2018). Recently, deep learning methods have been applied for this task (Ghosh and Veale, 2016; Tay et al., 2018). For excellent surveys on sarcasm and irony detection see (Wallace, 2015; Joshi et al., 2017).

However, when recognizing sarcastic intent even humans have difficulties sometimes when considering an utterance in isolation (Wallace et al., 2014). Recently an increasing number of researchers have started to explore the role of contextual information for irony and sarcasm analysis. The term context loosely refers to any *information* that is available beyond the utterance itself (Joshi et al., 2017). A few researchers have examined author context (Bamman and Smith, 2015; Khattri et al., 2015; Rajadesingan et al., 2015; Amir et al., 2016; Ghosh and Veale, 2017), multi-modal context (Schifanella et al., 2016; Cai et al., 2019; Castro et al., 2019), eye-tracking information (Mishra et al., 2016), or conversation context (Bamman and Smith, 2015; Wang et al., 2015; Joshi et al., 2016; Zhang et al., 2016; Ghosh et al., 2017; Ghosh and Veale, 2017).

Related to shared tasks on figurative language analysis, recently, Van Hee et al. (2018) have con-

ducted a SemEval task on irony detection in Twitter focusing on utterances in isolation. Besides the binary classification task of identifying the ironic tweet the authors also conducted a multi-class irony classification to identify the specific *type* of irony: whether it contains verbal irony, situational irony, or other types of irony. In our case, the current shared task aims to study the role of conversation context for sarcasm detection. In particular, we focus on benchmark the effectiveness of modeling the conversation context (e.g., all the prior dialogue turns or a subset of the prior dialogue turns) for sarcasm detection.

3 Task Description

The design of our shared task is guided by two specific issues. First, we plan to leverage a particular type of context — the entire prior conversation context — for sarcasm detection. Second, we plan to investigate the systems' performance on conversations from two types of social media platforms: Twitter and Reddit. Both of these platforms allow the writers to mark whether their messages are sarcastic (e.g., #sarcasm hashtag in Twitter and "/s" marker in Reddit).

The competition is organized in two phases: training and evaluation. By making available common datasets and frameworks for evaluation, we hope to contribute to the consolidation and strengthening of the growing community of researchers working on computational approaches to sarcasm analysis.

3.1 Datasets

3.1.1 Reddit Training Dataset

Khodak et al. (2017) introduced the self-annotated Reddit Corpus which is a very large collection of sarcastic and non-sarcastic posts (over one million) curated from different subreddits such as politics, religion, sports, technology, etc. This corpus contains self-labeled sarcastic posts where users label their posts as sarcastic by marking "/s" to the end of sarcastic posts. For any such sarcastic post, the corpus also provides the full conversation context, i.e., all the prior turns that took place in the dialogue.

We select the training data for the Reddit track from Khodak et al. (2017). We considered a couple of criteria. First, we choose sarcastic responses with at least two prior turns. Note, for many responses in our training corpus the number of turns is much more. Second, we curated sarcastic re-

sponses from a variety of subreddits such that no single subreddit (e.g., politics) dominates the training corpus. In addition, we avoid responses from subreddits that we believe are too specific and narrow (e.g., subreddit dedicated to a specific video game) that might not generalize well. The non-sarcastic partition of the training dataset is collected from the same set of subreddits that are used to collect sarcastic responses. We finally end up in selecting 4,400 posts (as well as their conversation context) for the training dataset equally balanced between sarcastic and non-sarcastic posts.

3.1.2 Twitter Training Dataset

For the Twitter dataset, we have relied upon the annotations that users assign to their tweets using hashtags. The sarcastic tweets were collected using hashtags: *#sarcasm* and *#sarcastic*. As non-sarcastic utterances, we consider sentiment tweets, i.e., we adopt the methodology proposed in related work (Muresan et al., 2016). Such sentiment tweets do not contain the sarcasm hashtags but include hashtags that contain positive or negative sentiment words. The positive tweets express direct positive sentiment and they are collected based on tweets with positive hashtags such as *#happy*, *#love*, *#lucky*. Likewise, the negative tweets express direct negative sentiment and are collected based on tweets with negative hashtags such as *#sad*, *#hate*, *#angry*. Classifying sarcastic utterances against sentiment utterances is a considerably harder task than classifying against random objective tweets since many sarcastic utterances also contain sentiment terms. Here, we are relying on *self-labeled* tweets, thus, it is always possible that sarcastic tweets were mislabeled with sentiment hashtags or users did not use the #sarcasm hashtag at all. We manually evaluated around 200 sentiment tweets and found very few such cases in the training corpus. Similar to the Reddit dataset we apply a couple of criteria while selecting the training dataset. First, we select sarcastic or non-sarcastic tweets only when they appear in a dialogue (i.e., begins with "@"-user symbol) and at least have two or more prior turns as conversation context. Second, for the non-sarcastic posts, we maintain a strict upper limit (i.e., not-greater than 10%) for any sentiment hashtag. Third, we apply heuristics such as avoiding short tweets, discarding tweets with only multiple URLs, etc. We end up selecting 5,000 tweets for training balanced between sarcastic and non-sarcastic tweets.
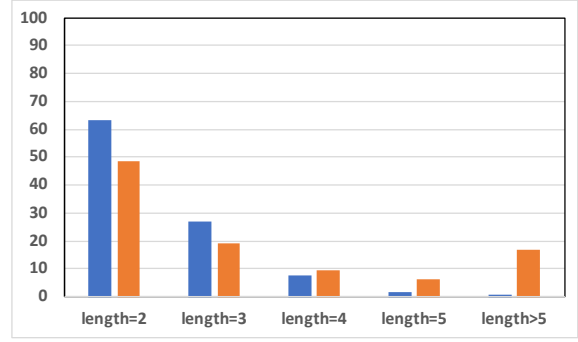

Figure 1: Plot of Reddit (blue) and Twitter (orange) training datasets on the basis of context length. X-axis represents context length (i.e., number of prior turns) and Y-axis represents the % of training utterances.

Figure 1 presents a plot of number of training utterances on the basis of context length, for Reddit and Twitter tracks respectively. We notice, although the numbers are comparable for utterances with context length equal to two or three, for Twitter corpus, utterances with a higher number of context (i.e., prior turns) is much higher.

3.1.3 Evaluation Data

The Twitter data for evaluation is curated similarly to the training data. For Reddit, we do not use Khodak et al. (2017) rather collected new sarcastic and non-sarcastic responses from Reddit. First, for sarcastic responses we utilize the same set of subreddits utilized in the training dataset, thus, keeping the same genre between the evaluation and training. For the non-sarcastic partition, we utilized the same set of subreddits and submission threads as the sarcastic partition. For both tracks the evaluation dataset contains 1800 instances partitioned equally between the sarcastic and the non-sarcastic categories.

3.2 Training Phase

In the first phase, data is released for training and/or development of sarcasm detection models (both Reddit and Twitter). Participants can choose to partition the training data further to a validation set for preliminary evaluations and/or tuning of hyper-parameters. Likewise, they can also elect to perform cross-validation on the training data.

3.3 Evaluation Phase

In the second phase, instances for evaluation are released. Each participating system generated predictions for the evaluation instances, for up to N

3

models. [1] Predictions are submitted to the CodaLab site and evaluated automatically against the gold labels. CodaLab is an established platform to organize shared-tasks (Leong et al., 2018) because it is easy to use, provides easy communication with the participants (e.g., allows mass-emailing) as well as tracks all the submissions updating the leaderboard in real-time. The metrics used for evaluation is the average F1 score between the two categories - sarcastic and non-sarcastic. The leaderboards displayed the Precision, Recall, and F1 scores in the descending order of the F1 scores, separately for the two tracks - Twitter and Reddit.

4 Systems

The shared task started on January 19, 2020, when the training data was made available to all the registered participants. We released the evaluation data on February 25, 2020. Submissions were accepted until March 16, 2020. Overall, we received an overwhelming number of submissions: 655 for the Reddit track and 1070 for the Twitter track. The CodaLab leaderboard showcases results from 39 systems for the Reddit track and 38 systems for the Twitter track, respectively. Out of all submissions, 14 shared task system papers were submitted. In the following section we summarize each system paper. We also put forward a comparative analysis based on their performance and the choice of features/models in Section 5. Interested readers can refer to the individual teams' papers for more details. But first, we discuss the baseline classification model that we used.

4.1 Baseline Classifier

We use prior published work as the baseline that used conversation context to detect sarcasm from social media platforms such as Twitter and Reddit (Ghosh et al., 2018). Ghosh et al. (2018) proposed a *dual LSTM architecture* with hierarchical attention where one LSTM models the conversation context and the other models sarcastic response. The hierarchical attention (Yang et al., 2016) implements two levels of attention – one at the word level and another at the sentence level. We used their system based on only the immediate conversation context (i.e., the immediate prior turn). [2] This is denoted as $LSTM_{attn}$ in Table 3 and Table 4.

---

[1]N is set to 999.
[2]https://github.com/Alex-Fabbri/deep_learning_nlp_sarcasm

4.2 System Descriptions

We describe the participating systems in the following section (in alphabetical order).

abaruah (Baruah et al., 2020): Fine-tuned a BERT model (Devlin et al., 2018) and reported results on varying maximum sequence length (corresponding to varying level of context inclusion from just response to entire context). They also reported results of BiLSTM with FastText embeddings (of response and entire context) and SVM based on char n-gram features (again on both response and entire context). One interesting result was SVM with discrete features performed better than BiLSTM. They achieved best results with BERT on response and most immediate context.

ad6398 (Kumar and Anand, 2020): Report results comparing multiple transformer architectures (BERT, SpanBERT (Joshi et al., 2020), RoBERTa (Liu et al., 2019)) both in single sentence classification (with concatenated context and response string) and sentence pair classification (with context and response being separate inputs to a Siamese type architecture). Their best result was with using RoBERTa + LSTM model.

aditya604 (Avvaru et al., 2020): Used BERT on simple concatenation of last-k context texts and response text. The authors included details of data cleaning (de-emojification, hashtag text extraction, apostrophe expansion) as well experiments on other architectures (LSTM, CNN, XLNet (Yang et al., 2019)) and varying size of context (5, 7, complete) in their report. The best results were obtained by BERT with 7 length context for Twitter dataset and BERT with 5 context for Reddit dataset.

amitjena40 (Jena et al., 2020): Used a time-series analysis inspired approach for integrating context. Each text in conversational thread (context and response) was individually scored using BERT and Simple Exponential Smoothing (SES) was utilized to get probability of final response being sarcastic. They used the final response label as a pseudo-label for scoring the context entries, which is not theoretically grounded. If final response is sarcastic, the previous context dialogue cannot be assumed to be sarcastic (with respect to its preceding dialogue). However, the effect of this error is attenuated due to exponentially decreasing contribution of context to final label under SES scheme.

Rank	Lb. Rank	Team	P	R	F1	Approach
1	1	miroblog	0.834	0.838	0.834	BERT + BiLSTM + NeXtVLAD + Context Ensemble + Data Augmentation
2	2	andy3223	0.751	0.755	0.750	RoBERTa-Large (all the prior turns)
3	6	taha	0.738	0.739	0.737	BERT+ Local Context Focus
4	8	tanvidadu	0.716	0.718	0.716	RoBERTa-Large (last two prior turns)
5	9	nclabj	0.708	0.708	0.708	RoBERTa + Multi-Initialization Ensemble
6	12	ad6398	0.693	0.699	0.691	RoBERTa + LSTM
7	16	kalaivani.A	0.679	0.679	0.679	BERT (isolated response)
8	17	amitjena40	0.679	0.683	0.678	TorchMoji + ELMO + Simple Exp. Smoothing
9	21	burtenshaw	0.67	0.677	0.667	Ensemble of SVM, LSTM, CNN-LSTM, MLP
10	26	salokr	0.641	0.643	0.639	BERT + CNN + LSTM
11	31	adithya604	0.605	0.607	0.603	BERT (concatenation of prior turns and response)
12	-	baseline	0.600	0.599	0.600	$LSTM_{attn}$
13	32	abaruah	0.595	0.605	0.585	BERT-Large (concatenation of response and its immediate prior turn)

Table 3: Performance of the best system per team and baseline for the Reddit track. We include two ranks - ranks from the submitted systems as well as the Leaderboard ranks from the CodaLab site

AnandKumaR (Khatri and P, 2020): Experimented with using traditional ML classifiers like SVM and Logisitic Regression over embeddings through BERT and GloVe (Pennington et al., 2014). Using BERT as a feature extraction method as opposed to fine-tuning it was not beneficial and Logisitic Regression over GloVe embeddings outperformed them in their experiment. Context was used in their best model but no details were available about the depth of context usage (full vs. immediate). Additionally, they only experimented with Twitter data and no submission was made to the Reddit track. They provided details of data cleaning measures for their experiments which involved stopword removal, lowercasing, stemming, punctuation removal and spelling normalization.

andy3223 (Dong et al., 2020): Used the transformer-based architecture for sarcasm detection, reporting the performance of three architecture, BERT, RoBERTa, and ALBERT (Lan et al., 2019). They considered two models, the *target-oriented* where only the target (i.e., sarcastic response) is modeled and *context-aware*, where the context is also modeled with the target. The authors conducted extensive hyper-parameter search, and set the learning rate to 3e-5, the number of epochs to 30, and use different seed values, 21, 42, 63, for three runs. Additionally, they set the maximum sequence length 128 for the *target-oriented* models while it is set to 256 for the *context-aware* models.

burtenshaw (Lemmens et al., 2020): Employed an ensemble of four models - LSTM (on word, emoji and hashtag representations), CNN-LSTM (on GloVe embeddings with discrete punctuation and sentiment features), MLP (on sentence embeddings through Infersent (Conneau et al., 2017)) and SVM (on character and stylometric features). The first three models (except SVM) used the last two immediate contexts along with the response.

duke_DS (Gregory et al., 2020): Here the authors have conducted extensive set of experiments using discrete features, DNNs, as well as transformer models, however, reporting only the results on the Twitter track. Regarding discrete features, one of novelties in their approach is including a *predictor* to identify whether the tweet is political or not, since many sarcastic tweets are on political topics. Regarding the models, the best performing model is an ensemble of five transformers: BERT-base-uncased, RoBERTa-base, XLNet-base-cased, RoBERTa-large, and ALBERT-base-v2.

kalaivani.A (kalaivani A and D, 2020): Compared traditional machine learning classifiers (e.g., Logistic Regression/Random Forest/XGBoost/Linear SVC/ Gaussian Naive Bayes) on discrete bag-of-word features/Doc2Vec features with LSTM models on Word2Vec embeddings (Mikolov et al., 2013) and BERT

models. For context usage they report results on using isolated response, isolated context and context-response combined (unclear as to how deep the context usage is). The best performance for their experiments was by BERT on isolated response.

miroblog (Lee et al., 2020): Implemented a classifier composed of BERT followed by BiLSTM and NeXtVLAD (Lin et al., 2018) (a differentiable pooling mechanism which empirically performed better than Mean/Max pooling). [3] They employed an ensembling approach for including varying length context and reported that gains in F1 after context of length three are negligible. Just with these two contributions alone, their model outperformed all others. Additionally, they devised a novel approach of data augmentation (i.e., Contextual Response Augmentation) from unlabelled conversational contexts based on next sentence prediction confidence score of BERT. Leveraging large-scale unlabelled conversation data from web, their model outperformed the second best system by 14% and 8.4% for Twitter and Reddit respectively (absolute F1 score).

nclabj (Jaiswal, 2020): Used a majority-voting ensemble of RoBERTa models with different weight-initialization and different levels of context length. Their report shows that previous 3 turns of dialogues had the best performance in isolation. Additionally, the present results comparing other sentence embedding architectures like Universal Sentence Encoder (Cer et al., 2018), ELMo (Peters et al., 2018) and BERT.

salokr/vaibhav (Srivastava et al., 2020) : Employed a CNN-LSTM based architecture on BERT embeddings to utilize the full context thread and the response. The entire context after encoding through BERT is passed through CNN and LSTM layers to get a representation of the context. Convolution and dense layers over this summarized context representation and BERT encoding of response make up the final classifier.

taha (ataei et al., 2020): Reported experiments comparing SVM on character n-gram features, LSTM-CNN models, Transformer models as well as a novel usage of aspect based sentiment classification approaches like Interactive Attention

Networks(IAN) (Ma et al., 2017), Local Context Focus(LCF)-BERT (Zeng et al., 2019) and BERT-Attentional Encoder network (AEN) (Song et al., 2019). For aspect based approaches, they viewed the last dialogue of conversational context as aspect of the target response. LCF-BERT was their best model for the Twitter task but due to computational resource limitations they were not able to try it for Reddit task (where BERT on just the response text performed best).

tanvidadu (Dadu and Pant, 2020): Fine-tuned RoBERTa-large model (355 Million parameters with over a 50K vocabulary size) on response and its two immediate contexts. They reported results on three different types of inputs: response-only model, concatenation of immediate two context with response, and using an explicit separator token between the response and the final context. The best result is reported in the setting where they used the separation token.

5 Results and Discussions

Table 3 and Table 4 present the results for the Reddit track and the Twitter track, respectively. We show the rank of the submitted systems (best result from their submitted reports) both in terms of the system submissions (out of 14) as well as their rank on the Codalab leaderboard. Note, for a couple of entries we observe a discrepancy between their best reported system(s) and the leaderboard entries. For the sake of fairness, for such cases, we selected the leaderboard entries to present in Table 3 and Table 4. [4]

Also, out of the 14 system descriptions *duke_DS* and *AnadKumR* report the performance on the Twitter dataset, only. For overall results on both tracks, we observe majority of the models outperformed the $LSTM_{attn}$ baseline (Ghosh et al., 2018). Almost all the submitted systems have used the transformer-architecture that seems to perform better than RNN-architecture, even without any task-specific fine-tuning. Although most of the models are similar and perform comparably, we observe a particular system - miroblog - has outperformed the other models in both the tracks by posting an improvement over the 2nd ranked system by more than 7% F1-score in the Reddit track and by 14% F1-score in the Twitter track.

---

[3] VLAD is an acronym of "Vector of Locally Aggregated Descriptors" (Lin et al., 2018).

[4] Also, for such cases (e.g., abaruah, under the *Approach* column we reported the approach described in the system paper that is not necessarily reflect the scores of Table 3.

Rank	Lb. Rank	Team	P	R	F1	Approach
1	1	miroblog	0.932	0.936	0.931	BERT + BiLSTM + NeXtVLAD + Context Ensemble + Data Augmentation
2	2	nclabj	0.792	0.793	0.791	RoBERTa + Multi-Initialization Ensemble
3	3	andy3223	0.791	0.794	0.790	RoBERTa-Large (all the prior turns)
4	5	ad6398	0.773	0.774	0.772	RoBERTa + LSTM
5	6	tanvidadu	0.772	0.772	0.772	RoBERTa-Large (last two prior turns)
6	8	duke_DS	0.758	0.767	0.756	Ensemble of Transformers
7	11	amitjena40	0.751	0.751	0.750	TorchMoji + ELMO + Simple Exp. Smoothing
8	13	salokr	0.742	0.746	0.741	BERT + CNN + LSTM
9	16	burtenshaw	0.741	0.746	0.740	Ensemble of SVM, LSTM, CNN-LSTM, MLP
10	21	abaruah	0.734	0.735	0.734	BERT-Large (concatenation of response and its immediate prior turn)
11	24	taha	0.731	0.732	0.731	BERT
12	27	kalaivani.A	0.722	0.722	0.722	BERT (isolated response)
13	28	adithya604	0.719	0.721	0.719	BERT (concatenation of prior turns and response)
14	35	AnadKumR	0.690	0.690	0.690	GloVe + Logistic Regression
15	-	baseline	0.700	0.669	0.680	$LSTM_{attn}$

Table 4: Performance of the best system per team and baseline for the Twitter track. We include two ranks - ranks from the submitted systems as well as the Leaderboard ranks from the CodaLab site

In the following paragraphs, we inspect the performance of the different systems more closely. We discuss a couple of particular aspects.

Context Usage: One of the prime motivating factors for conducting this shared task was to investigate the role of contextual information. We notice the most common approach for integrating context was simply concatenating it with the response text. Novel approaches include :

1. Taking immediate context as aspect for response in Aspect-based Sentiment Classification architectures (**taha**)

2. CNN-LSTM based summarization of entire context thread (**salokr**)

3. Time-series fusion with proxy labels for context (**amitjena40**)

4. Ensemble of multiple models with different depth of context (**miroblog**)

5. Using explicit separator between context and response when concatenating (**tanvidadu**)

Depth of Context: Results suggest that beyond three context turns, gains from context information are negligible and may also reduce the performance due to sparsity of long context threads. The depth of context required is dependent on the architecture and CNN-LSTM based summarization of context thread (**salokr**) was the only approach that effectively used the whole dialogue.

Discrete vs. Embedding Features The leaderboard was dominated by Transformer based architectures and we saw submissions using BERT or RoBERTa and other variants. Other sentence embedding architectures like Infersent, CNN/LSTM over word embeddings were also used but had middling performances. Discrete features were involved in only two submissions (**burtenshaw** and **duke_DS**) and were the focus of **burtenshaw** system.

Leveraging other datasets The large difference between the best model (**miroblog**) and other systems can be attributed to their dataset augmentation strategies. Using just the context thread as a negative example when the context+response is a positive example, is a straight-forward approach for augmentation from labeled dialogues. Their novel contribution lies in leveraging large-scaled unlabelled dialogue threads, showing another use of BERT by using NSP confidence score for assigning pseudo-labels.

Analysis of predictions: Finally, we conducted an error analysis based on the predictions of the

systems. We particularly focused on addressing two questions. First, we investigate whether any particular pattern exists in the evaluation instances that are wrongly classified by the majority of the systems. Second, we compare the predictions of the top-performing systems to identify instances correctly classified by the candidate system but missed by the remaining systems. Here, we attempt to recognize specific characteristics that are unique to a model, if any.

Instead of looking at the predictions of all the systems we decided to analyze only the *top-three* submissions in both tracks because of their high performances. We identify 80 instances (30 sarcastic) from the Reddit evaluation dataset and 20 instances (10 sarcastic) from the Twitter evaluation set, respectively, that are missed by all the top-performing systems. Our interpretation of this finding is that all these test instances more or less belong to a variety of topics including sarcastic remarks on baseball teams, internet bills, vaccination, etc., that probably do not generalize well during the training. For both Twitter and Reddit, we also found many sarcastic examples that contain common non-sarcastic markers such as laughs (e.g., "haha"), jokes, positive-sentiment emoticons (e.g., :)) in terms of Twitter track. We did not find any correlation to context length. Most of the instances contain varied context length, from two to six.

While analyzing the predictions of individual systems we noted that **miroblog** correctly identifies the most number of predictions for both the tracks. In fact, miroblog has successfully predicted over two hundred examples (with almost equal distribution of sarcastic and non-sarcastic instances) in comparison to the second-ranked and third-ranked systems for both tracks. As stated earlier, this can be attributed to their data augmentation strategies that have assisted miroblog's models to generalize best. However, we still notice that instances with subtle humor or positive sentiment are missed by the best-performing models even if they are pre-trained on a very large-scale corpora. We foresee models that are able to detect subtle humor or witty wordplay will perform even better in a sarcasm detection task.

6 Conclusion

This paper summarizes the results of the shared task on sarcasm detection using conversation from two social media platforms (Reddit and Twitter),

organized as part of the 2nd Workshop on the Figurative Language Processing at ACL 2020. This shared task aimed to investigate the role of conversation context for sarcasm detection. The goal was to understand how much conversation context is needed or helpful for sarcasm detection. For Reddit, the training data was sampled from the standard corpus from Khodak et al. (2017) whereas we curated a new evaluation dataset. For Twitter, both the training and the test datasets are new and collected using standard hashtags. We received 655 submissions (from 39 unique participants) and 1070 submissions (from 38 unique participants) for Reddit and Twitter tracks, respectively. We provided brief descriptions of each of the participating systems who submitted a shared task paper (14 systems).

We notice that almost every submitted system have used transformer-based architectures, such as BERT and RoBERTa and other variants, emphasizing the increasing popularity of using pre-trained language models for various classification tasks. The best systems, however, have employed a clever mix of ensemble techniques and/or data augmentation setups, which seem to be a promising direction for future work. We hope that some of the teams will make their implementations publicly available, which would facilitate further research on improving performance on the sarcasm detection task.

References

kalaivani A and Thenmozhi D. 2020. Sarcasm identification and detection in conversion context using BERT. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA.*

Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976.*

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, Seattle, WA, USA.*

Adithya Avvaru, Sanath Vobilisetty, and Radhika Mamidi. 2020. Detecting sarcasm in conversation context using Transformer based model. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA.*

David Bamman and Noah A Smith. 2015. Contextual-

ized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media*.

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, Seattle, WA, USA*.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multimodal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515.

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.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Tanvi Dadu and Kartikey Pant. 2020. Sarcasm detection using context separators in online discourse. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA*.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10.

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

Xiangjue Dong, Changmao Li, and Jinho D. Choi. 2020. Transformer-based context-aware sarcasm detection in conversation threads from social media. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA*.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625. Association for Computational Linguistics.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of NAACL-HLT*, pages 161–169.

Aniruddha Ghosh and Tony Veale. 2017. Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 482–491. Association for Computational Linguistics.

Debanjan Ghosh, Alexander R Fabbri, and Smaranda Muresan. 2018. Sarcasm analysis using conversation context. *arXiv preprint arXiv:1808.07531*.

Debanjan Ghosh, Alexander Richard Fabbri, and Smaranda Muresan. 2017. The role of conversation context for sarcasm detection in online interactions. *arXiv preprint arXiv:1707.06226*.

Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. 2015. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1003–1012, Lisbon, Portugal. Association for Computational Linguistics.

Debanjan Ghosh and Smaranda Muresan. 2018. " with 1 follower i must be awesome: P". exploring the role of irony markers in irony recognition. *arXiv preprint arXiv:1804.05253*.

Debanjan Ghosh, Elena Musi, Kartikeya Upasani, and Smaranda Muresan. 2019. Interpreting verbal irony: Linguistic strategies and the connection to the type of semantic incongruity. *arXiv preprint arXiv:1911.00891*.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *ACL (Short Papers)*, pages 581–586. Association for Computational Linguistics.

Hunter Gregory, Steven Li, Pouya Mohammadi, Natalie Tarn, Rachel Ballantyne, and Cynthia Rudin. 2020. A Transformer approach to contextual sarcasm detection in twitter. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA*.

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. Association for Computational Linguistics.

Nikhil Jaiswal. 2020. Neural sarcasm detection using conversation context. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA*.

Amit Kumar Jena, Aman Sinha, and Rohit Agarwal. 2020. C-net: Contextual network for sarcasm detection. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA*.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, page 73.

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

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, and Mark Carman. 2016. Harnessing sequence labeling for sarcasm detection in dialogue from tv series 'friends'. *CoNLL 2016*, page 146.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Akshay Khatri and Pranav P. 2020. Sarcasm detection in tweets with BERT and GloVe embeddings. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA*.

Anupam Khattri, Aditya Joshi, Pushpak Bhattacharyya, and Mark Carman. 2015. Your sentiment precedes you: Using an author's historical tweets to predict sarcasm. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 25–30, Lisboa, Portugal. Association for Computational Linguistics.

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

Amardeep Kumar and Vivek Anand. 2020. Transformers on sarcasm detection with context. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Hankyol Lee, Youngjae Yu, and Gunhee Kim. 2020. Augmenting data for sarcasm detection with unlabeled conversation context. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA*.

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, Seattle, WA, USA*.

Chee Wee Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 vua metaphor detection shared task. In *Proceedings of the Workshop on Figurative Language Processing*, pages 56–66.

CC Liebrecht, FA Kunneman, and APJ van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets# not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.

Rongcheng Lin, Jing Xiao, and Jianping Fan. 2018. Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0.

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 preprint arXiv:1907.11692*.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. *arXiv preprint arXiv:1709.00893*.

Navonil Majumder, Soujanya Poria, Haiyun Peng, Niyati Chhaya, Erik Cambria, and Alexander Gelbukh. 2019. Sentiment and sarcasm classification with multitask learning. *IEEE Intelligent Systems*, 34(3):38–43.

Diana Maynard and Mark A Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of LREC*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016. Harnessing cognitive features for sarcasm detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1095–1104, Berlin, Germany.

Smaranda Muresan, Roberto Gonzalez-Ibanez, Debanjan Ghosh, and Nina Wacholder. 2016. Identification of nonliteral language in social media: A case study on sarcasm. *Journal of the Association for Information Science and Technology*, 67(11):2725–2737.

Silviu Oprea and Walid Magdy. 2019. Exploring author context for detecting intended vs perceived sarcasm. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2859.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 97–106. ACM.

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.

Rossano Schifanella, Paloma de Juan, Joel Tetreault, and Liangliang Cao. 2016. Detecting sarcasm in multimodal social platforms. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1136–1145. ACM.

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

Himani Srivastava, Vaibhav Varshney, Surabhi Kumari, and Saurabh Srivastava. 2020. A novel hierarchical BERT architecture for sarcasm detection. In *Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA*.

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

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*.

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.

Tony Veale and Yanfen Hao. 2010. Detecting ironic intent in creative comparisons. In *European Conference on Artificial Intelligence*, volume 215, pages 765–770, Lisbon, Portugal.

Byron C Wallace. 2015. Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, 43(4):467–483.

Byron C Wallace, Do Kook Choe, Laura Kertz, and Eugene Charniak. 2014. Humans require context to infer ironic intent (so computers probably do, too). In *ACL (2)*, pages 512–516.

Zelin Wang, Zhijian Wu, Ruimin Wang, and Yafeng Ren. 2015. Twitter sarcasm detection exploiting a context-based model. In *International Conference on Web Information Systems Engineering*, pages 77–91, Miami, Florida. Springer.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489.

Biqing Zeng, Heng Yang, Ruyang Xu, Wu Zhou, and Xuli Han. 2019. Lcf: A local context focus mechanism for aspect-based sentiment classification. *Applied Sciences*, 9(16):3389.

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