Sarcasm Detection in a Disaster Context

Tiberiu Sosea♣, Junyi Jessy Li♢, Cornelia Caragea♣
{tsosea2,cornelia}@uic.edu, jessy@austin.utexas.edu
♣Computer Science, University of Illinois Chicago
♢Linguistics, The University of Texas at Austin

Abstract

During natural disasters, people often use social media platforms such as Twitter to ask for help, to provide information about the disaster situation, or to express contempt about the unfolding event or public policies and guidelines. This contempt is in some cases expressed as sarcasm or irony. Understanding this form of speech in a disaster-centric context is essential to improving natural language understanding of disaster-related tweets. In this paper, we introduce HurricaneSARC, a dataset of $15,000$ tweets annotated for intended sarcasm, and provide a comprehensive investigation of sarcasm detection using pre-trained language models. Our best model is able to obtain as much as 0.70 F1 on our dataset. We also demonstrate that the performance on HurricaneSARC can be improved by leveraging intermediate task transfer learning. We release our data and code at https://github.com/tsosea2/HurricaneSarc.

Keywords: sarcasm detection, social media

1. Introduction

Understanding sarcasm from text is crucial for enabling progress on natural language understanding. However, despite being widely researched as an NLP task (Riloff et al., 2013; Wallace et al., 2014; Joshi et al., 2015; Amir et al., 2016; Oraby et al., 2016; Hazarika et al., 2018; Oprea and Magdy, 2020), to date, sarcasm detection was only explored in a general domain (e.g., the general Twitter or general Reddit) with no focus on a specific context, e.g., a disaster context.

Natural disasters such as hurricanes and earthquakes cause substantial material destruction and emotional damage to millions of people every year (Ritchie and Roser, 2020), with many being uprooted, evacuated, or disrupted from their daily activities. During disasters, affected individuals often turn to social media platforms such as Twitter and Facebook to look for updates, to post requests for help, to share their feelings, and to express contempt towards the unfolding event or public policies and guidelines. This contempt is in some cases expressed as the sophisticated linguistic phenomenon that makes use of figurative language: the sarcasm (or irony). Understanding this form of speech in a disaster-centric context is essential to improving the understanding of disaster-related tweets and their intended semantic meaning. However, detecting sarcasm solely from disaster-related short tweets when no visual or acoustic information is available is very challenging because of the difficulty of the task itself even for humans (in the absence of the writer's intent); the lack of sufficient textual context to leverage on; and the lack of annotated datasets for this task.

To this end, we explore the difficulty of sarcasm detection in disaster-related tweets and present HurricaneSARC, a dataset that contains $15,000$ tweets annotated with *sarcasm* and *non-sarcasm* using crowdsourcing. Unlike existing sarcasm datasets that cover a general domain, to our knowledge, HurricaneSARC is the first dataset labeled for sarcasm that covers a specific domain, i.e., tweets sampled from a disaster-centric domain—Hurricanes Harvey, Irma, and Maria, with the goal to measure progress on sarcasm detection in a focused context. Table 1 shows examples of sarcastic and non-sarcastic tweets from HurricaneSARC. As we can see, in the tweet *Are you having a good laugh in your dry, safe, comfortable home while #Harvey destroys lives.*, the writer is expressing sarcasm implicitly by using conceptually opposite phrases, such as *good laugh*, *safe*, and *Harvey destroys lives*. On the other hand, in the tweet *Sending good vibes to people in Puerto Rico!! stay safe.*, the writer is expressing sympathy towards the individuals affected by the disaster, hence it contains no sarcastic information.

Using HurricaneSARC, we contrast BERTweet (Nguyen et al., 2020), a pre-trained language model that learns effective language representations from unlabeled Twitter data, with Convolutional Neural Networks (Kim, 2014a) and BiLSTM (Hochreiter and Schmidhuber, 1997) to establish a baseline on our dataset. Moreover, we improve upon this baseline result, by leveraging intermediate task pretraining (Pruksachatkun et al., 2020) to investigate information transfers between sarcasm detection and related tasks such as emotion and sentiment analysis.

As disasters strike, large amounts of user-generated content are produced on social sites. However, due to the nature of disasters unfolding

LREC-COLING 2024, pages 14313–14324
20-25 May, 2024.

SARCASTIC	Are you having a good laugh in your dry, safe, comfortable home while #Harvey destroys lives.
SARCASTIC	We know natural disasters don't discriminate, but relief and who is most severely impacted tend to do so. #Harvey.
SARCASTIC	#Trump is excited by 125 mph winds of #Hurricane #Harvey in the way that a child is excited by his toys! Disgusting.
NOT SARCASTIC	Sending good vibes to people in Puerto Rico!! stay safe.
NOT SARCASTIC	#Harvey made landfall in Tex. at 11p as first major (Cat 3+) hurricane since Wilma in 2005.
NOT SARCASTIC	Maria, although downgraded to a category 4, is still a monster of a storm tearing through Puerto Rico right now.

Table 1: Examples from our dataset.

rapidly and the high costs needed for annotation, only a small quantity can be annotated and used for supervised classification. To address these issues, in this paper we investigate and contrast two potential solutions. First, we carry out experiments to study models' understanding of sarcasm in disaster-related tweets when trained on existing datasets for sarcasm from the general domain (direct transfer). Second, we propose semi-supervised learning (SSL) to exploit the readily available unlabeled data together with small labeled data. Specifically, we explore *Uncertainty-aware Self-Training (UST)* (Mukherjee and Awadallah, 2020) and *Noisy Student Training* (Xie et al., 2020) using noising techniques such as *back translation* (Sennrich et al., 2015), and show that unlabeled data can improve the performance of our models significantly. Concretely, our BERTweet-based *UST* model that uses only 200 labeled examples performs within 1% F1 of the vanilla BERTweet model, which is trained on four times more data. In contrast, transfering information from domains such as Twitter or Reddit is not as effective as our SSL methods, decreasing the overall F1 by 15%.

We summarize our contributions as follows: 1) We introduce HURRICANESARC, a new dataset for sarcasm detection in the disaster domain and show that models trained on a general domain struggle on sarcasm detection when evaluated on a disaster domain; 2) We develop strong baselines on our dataset, and explore intermediate task transfer learning as a means to improve the performance on our dataset; 3) We propose semi-supervised learning as a means to obtain better language representations for time-critical events such as disasters.

2. Related Work

Due to its highly figurative nature, identifying sarcasm from textual data is defined as one of the most challenging tasks in NLP (Wallace et al., 2014), which has attracted significant attention in recent years. The approaches proposed range from rule-based methods (Veale and Hao, 2010; Maynard and Greenwood, 2014; Bharti et al., 2015; Riloff et al., 2013) and statistical approaches (Joshi et al., 2015; Tepperman et al., 2006; Kreuz and Caucci, 2007; Reyes and Rosso, 2012; Liebrecht et al., 2013; Wallace et al., 2015; Lukin and Walker, 2013; Oraby et al., 2016), to deep learning techniques (Joshi et al., 2016; Amir et al., 2016; Ghosh and Veale, 2016; Hazarika et al., 2018; Majumder et al., 2019; Savini and Caragea, 2020, 2022).

Maynard and Greenwood (2014) argued that the sentiment hashtags can be key indicators for the expression of sarcasm, and developed rule-based classifiers, aimed at identifying negative phrases in positive sentences, as well as mining hyperboles, which are usually great indicators of sarcasm. Reyes and Rosso (2012) used Naïve Bayes and Decision Trees to identify sarcasm in tweets based on features such as the presence of irony or humor. Other studies focused on recognizing interjections, punctuation symbols, intensifiers, hyperboles (Kreuz and Caucci, 2007), emoticons (Carvalho et al., 2009), exclamations (Tsur et al., 2010), and hashtags (Davidov et al., 2010) in sarcastic comments. On the other hand, Ghosh and Veale (2016) proposed a concatenation of Convolutional Neural Network, Long-Short Term Memory Network, and Deep Neural Network (CNN-LSTM-DNN) that outperformed many state-of-the-art statistical methods based on text features. Poria et al. (2016) developed a framework based on a pre-trained CNN to retrieve sentiment, emotion, and personality features for sarcasm recognition. Zhang et al. (2016) used a bi-directional gated recurrent neural network with a pooling mechanism to automatically detect features from tweets, as well as context information from the history of the author.

Majumder et al. (2019) developed a multi-task learning framework and applied it on a dataset of $1,000$ sentences annotated with both sarcasm and sentiment tags, introduced by Mishra et al. (2016). Their framework is based on a GRU (Chung et al., 2014) model, and fuses the sarcasm and sentiment-specific vectors through a tensor network.

In addition to developing approaches for identifying sarcasm, researchers also focused on creating datasets from different online platforms. SARC (Khodak et al., 2017) is a general Reddit Corpus en-

compassing 1.3 million sarcastic statements. The dataset is self-annotated (i.e., the sarcasm label is assigned by the author of the statement). iSarcasm (Oprea and Magdy, 2020) is a dataset of intended sarcasm from the general Twitter domain, which consists of 4,484 tweets manually labeled with the *sarcastic* and *non-sarcastic* categories. Several other datasets exist for sarcasm detection from platforms such as general Twitter and Reddit that are annotated using crowdsourcing (Riloff et al., 2013; Maynard and Greenwood, 2014; Ptáček et al., 2014; Oraby et al., 2016), or leveraging platform specific cues such as hashtags in Twitter (Davidov et al., 2010; González-Ibánez et al., 2011; Reyes et al., 2013). For example, Riloff et al. (2013) annotated $1,600$ tweets from the general Twitter domain with the *sarcastic* and *non-sarcastic* categories using crowdsourcing. On the other hand, Davidov et al. (2010) used hashtags such as #Sarcastic or #Sarcasm to create a corpus composed of 900 examples annotated with three categories: *positive*, *negative*, and *sarcastic*.

In contrast, HurricaneSARC contains tweets streamed during natural disaster events. To our knowledge, no previous dataset for sarcasm detection covers a specialized domain, e.g., a disaster domain. Moreover, composed of $15,000$ tweets manually annotated for *sarcasm*, our dataset enables complex exploration of pre-trained language models such as BERT (Devlin et al., 2019), as well as the study of domain gaps between our disaster setting and other general domains. We hope that HurricaneSARC will spur further research and offer a better understanding of time-critical and crucial events such as disasters.

3. Dataset

In this section, first we detail the construction of HurricaneSARC, followed by our anlysis into the lexical particularities of our dataset.

3.1. Dataset Construction

HurricaneSARC is a dataset of English tweets from the disaster domain, annotated for sarcasm. We randomly sampled $15,000$ tweets from the large-scale repository of tweets introduced by Ray Chowdhury et al. (2019), that were streamed during the Hurricanes Irma, Harvey, and Maria. Next, we labeled these tweets using the Amazon Mechanical Turk crowdsourcing platform. The Turk workers were provided definitions of sarcasm from dictionary.com and Wikipedia and were asked to annotate tweets for the expression of sarcasm, i.e., to assign the *true* label for tweets perceived as containing sarcasm (or irony), and *false* otherwise. The annotators were paid fairly. Each tweet was annotated by five different workers, and the final label for a tweet was computed using majority vote. To ensure qualitative results, we employed strict annotator requirements for the task, such as high acceptance rate ($>95\%$), and a large amount of completed tasks ($500+$).

The annotation process produced 820 sarcastic tweets, which amount for 5.5% of the total number of sampled tweets. These tweets represent positive samples (i.e., tweets labeled as expressing sarcasm). To create the negative samples, we sampled an equal amount from the remaining pool (i.e., the tweets labeled as not expressing sarcasm). We also experimented with all negative samples as opposed to downsampling an equal amount. However, this significantly hurt model performance. Thus, consistent with Khodak et al. (2017), we construct a balanced version of the dataset on which we report the modeling results in the following sections. However, we make both dataset versions publicly available to enable further progress on sarcasm detection and understanding of sarcasm in a focused context. In our final balanced version of the dataset, we use $\sim 25\%$ of the data for testing, $\sim 18\%$ for validation, and the rest for training. Table 2 shows the number of examples in each split.

Set	Sarcastic	Non-Sarcastic	Total
Train	467	467	934
Validation	150	150	300
Test	200	200	400

Table 2: HurricaneSARC benchmark dataset size.

3.2. Lexical Analysis

We perform a word level lexical analysis to gain insights into the vocabulary used in HurricaneSARC (compared with a general domain) and explore potential challenges and particularities of our dataset. First, we investigate the occurence of emotion and sentiment intensive words in sarcastic and non-sarcastic tweets.

To this end, we use EmoLex (Mohammad and Turney, 2013), a lexicon containing words and the associated emotion and sentiment evoked by these words. We study this co-occurence on HurricaneSarc by contrast with the iSarcasm dataset introduced by Oprea and Magdy (2020) from a general domain and report the co-occurence ratios in Table 3. Note that both datasets are compiled from Twitter. First, we observe that only 18% of the sarcastic tweets in HurricaneSarc contain words expressing joy, whereas the ratio of non-sarcastic tweets containing joy words is significantly larger, up to 34%. In contrast, we observe a different pattern in the iSarcasm dataset, where both the sarcastic

	POS	NEG	JOY	FER	SDN	DSG	ANG	ANT	SRP
SARCASTIC-HS	0.48	0.70	0.18	0.57	0.28	0.22	0.32	0.23	0.25
NON-SARCASTIC-HS	0.41	0.58	0.34	0.52	0.20	0.13	0.26	0.22	0.20
SARCASTIC-IS	0.53	0.35	0.34	0.21	0.18	0.12	0.19	0.35	0.19
NON-SARCASTIC-IS	0.49	0.33	0.32	0.19	0.20	0.15	0.18	0.33	0.18

Table 3: Co-occurence of sarcastic/non-sarcastic tweets from HURRICANESARC (HS) and **iSarcasm** (IS) with words expressing a positive (POS) or negative (NEG) sentiment, as well as words expressing emotions such as joy (JOY), fear (FER), sadness (SDN), disgust (DSG) and anger (ANG), anticipation (ANT) and surprise (SRP). The numbers are normalized by the number of tweets in the sarcastic or non-sarcastic class.

HurricaneSARC	My neighbor is giving hurricane tips on Facebook meanwhile he already evacuated and left his yard full of projectiles.
HurricaneSARC	Holy Christ, you insufferable git. How about you help Puerto Rico? They were directly hit by a hurricane. Remember?. #Harvey.
iSarcasm	talents: making a joke of myself by crying in front of class.
iSarcasm	"healthcare is not a right"... Ok

Table 4: Examples from the HURRICANESARC and **iSarcasm** dataset.

Dataset	Sarcastic	Non-Sarcastic
HurricaneSARC	23.1	19.3
iSarcasm	17.2	18.5

Table 5: Average number of words per tweet for sarcastic/non-sarcastic comments.

and non-sarcastic tweets contain a similar ratio of words expressing joy. On the same note, the HURRICANESARC dataset has a considerable amount of negatively polarized words. In fact, as many as 70% of the tweets in HURRICANESARC contain at least a negative word, whereas in iSarcasm, this ratio is halved at 35%.

These findings suggest that the disaster-related tweets tend to be more emotion and sentiment-intensive than tweets from the general Twitter domain, since there are higher stakes involved in disaster scenarios. To better understand the particularities of our HURRICANESARC dataset, we compare a few representative examples from our dataset with some representative examples from the iSarcasm dataset in Table 4. We observe that HURRICANESARC is focused on the event and sarcasm is conveyed towards the ongoing disaster, while iSarcasm is much broader, containing tweets where the sarcasm is expressed towards various targets such as self-deprecation or healthcare.

We also note differences in the average number of words per tweet in our disaster scenario compared to the general Twitter domain. We show our findings in Table 5, where we compare our disaster setting with the general Twitter domain from the iSarcasm dataset (Oprea and Magdy, 2020). Interestingly, our tweets are longer in length than previous datasets, with the sarcastic tweets containing six additional words on average compared to iSarcasm.

4. Baseline Modeling

We model the sarcasm detection on HURRICANESARC using various methods:

Traditional Neural Methods We experiment with **(1) CNN** for text classification introduced by Kim (2014b) and **(2) Bi-LSTM** (Hochreiter and Schmidhuber, 1997) using pre-trained GloVe (Pennington et al., 2014) word embeddings as inputs.

Pre-trained Language Models We use the BERTweet (Nguyen et al., 2020) large-scale pre-trained language model for English Tweets, on top of which we add a linear layer for classification. We also experiment with the vanilla BERT model (Devlin et al., 2019).

4.1. Experimental Setting

All models are implemented using the Huggingface Transformers (Wolf et al., 2020) library. For training, we use a single Nvidia V100 GPU. We detail the best hyperparameters used and the computational resources needed to train our models in Appendix A. To ensure statistically significant results, we run the experiments 5 times, with different model weight initialization, and report the average of the results obtained from the five independent runs.

4.2. Results

We show the results using the above baselines in Table 6. First, we observe that CNN performs slightly better than BiLSTM. Second, we notice that the BERTweet model pre-trained on a Twitter corpus outperforms the vanilla BERT model trained

METHOD	P	R	F1
BI-LSTM	0.595	0.610	0.600
CNN	0.625	0.613	0.613
BERT	0.665	0.702	0.685
BERTWEET	**0.692**	**0.713**	**0.702**

Table 6: Precision (P), recall (R), F1-score (F1) on HURRICANESARC using neural models. We assert significance† if $p < 0.05$ under a paired-t test.

on Wikipedia and Bookcorpus by 1.7% in F1, and the other baselines by as much as 9% in F1. These results show that pre-trained language models are successful in learning useful language representations that are beneficial for the detection of sarcasm, and platform-specific pre-training (BERTweet pre-training) further improves the performance (compared with vanilla BERT).

Overall, despite that BERTweet performs better than the other baselines, its performance (in-domain) is still low, with the best method achieving only 0.702 F1 (Table 6). Next, we investigate various ways to introduce inductive biases into our models by performing a thorough investigation into intermediate task transfer learning.

5. Intermediate-Task Transfer Learning

To improve the performance of our baseline BERTweet model, we carry out a comprehensive set of experiments using intermediate task transfer learning (Han and Eisenstein, 2019; Pruksachatkun et al., 2020; Gururangan et al., 2020) (i.e., adjusting the contextualized embeddings to a target domain). Our framework is composed of two steps: First, we pre-train the BERTweet model on an intermediate supervised or unsupervised task. In the supervised setting, we add a task-specific classification layer, then train the model using a supervised loss on the intermediate task. On the other hand, in the unsupervised setting, we pre-train on unlabeled corpus using the dynamic masked language modeling objective (MLM). Second, after we train our models on an intermediate task, we remove any intermediate-task classification layers and initialize a new linear layer on top of BERTweet for training the model on HURRICANESARC.

5.1. Supervised Pre-training

We explore five intermediate tasks (described below) for the supervised scenario: 1) *EmoNet* is a dataset (Abdul-Mageed and Ungar, 2017) for fine-grained emotion detection in the general Twitter domain. It contains 1.6 million tweets labeled with the Plutchik-8 basic emotion categories (Plutchik, 1980). Here, we use a smaller version of the dataset containing $50,000$ examples provided by the authors; 2) *EmoNetSent* uses the same data as *EmoNet*. However, we aggregate the emotion labels based on the positive, negative, and neutral polarities; 3) *GoEmotions* (Demszky et al., 2020) is a dataset of $58,000$ Reddit comments annotated with 27 types of emotions; 4) *CARER* (Saravia et al., 2018) contains 664,462 English tweets from the general Twitter domain annotated with 8 emotion categories. To ensure a fair comparison with the previous datasets, we downsample the data to $50,000$ examples and use only these in our intermediate pre-training step; 5) *SARC* (Khodak et al., 2017) and 6) *iSarcasm* (Oprea and Magdy, 2020) are the datasets presented earlier in the paper.

5.2. Unsupervised Pre-training

For unsupervised intermediate tasks, we pre-train on dynamic masked language modeling (MLM) on three corpora: 1) *Hurricane* (Ray Chowdhury et al., 2019) is a large-scale repository of 15M tweets streamed during the Hurricanes Irma, Harvey, and Maria; 2) *EmoNet* (Abdul-Mageed and Ungar, 2017) is the emotion corpus from Twitter presented earlier. However, we remove the labels and train only on the tweets; 3) *SARC* is the sarcasm Reddit corpus presented earlier. Similar to *EmoNet*, we remove the labels and train only on the comments. The above datasets are summarized in Table 7.

Results Our intermediate task pre-training experiment results, shown in Table 12, reveal interesting details: First, supervised pre-training on the *EmoNet* emotion detection dataset is able to improve the performance over the simple BERTweet model by as much as 3% in F1. Meanwhile, we also notice a 1% improvement when using the *iSarcasm* dataset, as well as a 1% improvement using the *Carer* task. All these tasks contain data from the Twitter domain, which indicates that positive transfers of information tend to occur when transferring information between two related domains. In contrast, we observe negative transfers of information between *SARC* or *GoEmotions* and HURRICANESARC. We attribute the performance decrease to the disparity between the domains and platforms, i.e., both *SARC* and *GoEmotions* are collected from the Reddit platform, whereas our target domain is Twitter. In the unsupervised settings, performing MLM on HURRICANE yields the best performance improvement of 2% in F1. Interestingly, unlike the supervised scenario, pre-training on *SARC* is able to improve the performance over the vanilla BERTweet model, while *EmoNet* decreases the F1 score. However, both HURRICANE and *SARC* under-

DOMAIN	CARER	EMONET	ISARCASM	GOEMOTIONS	SARC	EMONET	SARC	HURRICANE-EXT
	SUPERVISED					UNSUPERVISED		
No. examples	$50,000$	$50,000$	$4,484$	$58,000$	$50,000$	$50,000$	$1.3M$	$15M$

Table 7: Number of examples in each intermediate task.

METHOD	P	R	F1
BERTWEET	0.692	0.713	0.702
CARER	0.715	0.712	0.713
EMONET	0.685	0.793	$\mathbf{0.732}^{\dagger}$
EMONET SENT	0.684	0.653	0.662
ISARCASM	0.645	$\mathbf{0.803}$	0.712
GOEMOTIONS	0.632	0.703	0.665
SARC	0.652	0.713	0.687
EMONET	0.652	0.683	0.675
SARC	0.684	0.707	0.698
HURRICANE-EXT	0.714	0.731	$0.723^{\dagger}$

Table 8: Precision, recall, and F1 using the following methods: **1)** Supervised Intermediate Pre-training using BERTweet (upper block). **2)** Unsupervised Pre-training using BERTweet (lower block). We assert significance† if $p < 0.05$ under a paired-t test with vanilla BERTweet.

went considerably more pre-training than *EmoNet*, due to the significantly smaller size of *EmoNet*.

We summarize our results as follows: 1) Both emotion detection and sarcasm detection make good supervised intermediate tasks for sarcasm detection, as long as the data used originates from the same social platform; 2) Additional unsupervised pretraining using masked language modeling helps the performance even when the intermediate data differs from the target data, provided that the tasks are closely related (e.g., sarcasm domain) and the training data has a considerable size. We also provide results of the experiments using the vanilla BERT model in Appendix B.

Error Analysis To investigate the effect of intermediate task fine-tuning, we perform an error analysis of our models. Specifically, we pass the examples from the test set through two models. First, we get the predictions of the vanilla BERTweet model, which underwent no additional pre-training. Second, we compute the predictions of the best performing emotion-aware BERTweet model, which underwent additional supervised pre-training on the *EmoNet* emotion dataset. Finally, we analyze examples where additional pre-training is able to change incorrect predictions into correct predictions, or correct predictions into incorrect ones. Our *EmoNet* BERTweet misclassifies 4 examples that are correctly classified by the vanilla BERTweet model. On the other hand, there are 28 examples misclassified by the simple BERTweet model, which are correctly classified by the *EmoNet* BERTweet model. We show a few of these examples in Table 9. Interestingly, these examples convey strong emotional messages. For example, in *People are dying in life support in Puerto Rico and the world is still in a frenzy about football.*, sadness and anger are strongly expressed along sarcasm. We argue that the supervised pre-training on *EmoNet* is able to induce important information about the emotions in a tweet, which correlate with the expressed sarcasm.

SARCASTIC	So you are happy about the death and destruction about to be done in Florida by IRMA ? I hope one of those.
SARCASTIC	People are dying in life support in Puerto Rico and the world is still in a frenzy about football.
NOT SARCASTIC	Maria-ravaged Puerto Rico left in misery without power, water and food
NOT SARCASTIC	If you're angry about the NFL then you need to get over yourselves because 3.5 MILLION AMERICANS are at risk of DEATH in Puerto Rico

Table 9: Examples corrected by the *EmoNet* BERTweet model.

Train \ Test	SARC	iSarcasm	HurricaneSARC
SARC	0.761	0.342	0.513
iSarcasm	0.672	0.391	0.552
HurricaneSARC	0.551	0.286	0.702

Table 10: F-1 scores of BERTweet when trained and tested on different sarcasm datasets.

6. Dealing with Time-Critical Events

As disasters start to unfold and unlabeled data starts to accumulate rapidly, annotating large amounts of tweets for sarcarm becomes very challenging. The time-critical nature of disasters and the annotation costs are two of the main challenges to understanding an ongoing disaster. In this sec-

METHOD	P	R	F1
BERTWEET-100	0.591	0.582	0.589 ± 2.7
BERTWEET-200	0.651	0.633	0.642 ± 2.0
BERTWEET-ALL	0.692	0.713	0.702 ± 0.94
NOISY STUDENT-100	0.628	0.612	0.620 ± 2.3
NOISY STUDENT-200	0.671	0.673	0.672 ± 1.7
NOISY STUDENT-ALL	0.737	0.735	$0.736^{\dagger} \pm 1.2$
UST-100	0.633	0.622	0.631 ± 2.2
UST-200	0.691	0.686	$0.689^{\dagger} \pm 1.7$
UST-ALL	**0.759**	**0.752**	$\mathbf{0.755 \pm 0.69}^{\dagger}$
CHATGPT	0.582	0.622	0.603

Table 11: Average precision, recall, and F-1 (averaged across the 5 runs) using semi-supervised learning and the vanilla BERTweet model. The number following the model name indicates the number of examples that model was trained on (e.g., BT-30 is the vanilla BERTweet model trained on 30 labeled examples). We assert significance† if $p < 0.05$ under a t-test with the counterpart vanilla BERTweet model (e.g., UST-200 vs. BT-200).

tion, we investigate various methods to overcome these challenges. We ask the following questions: Does one need annotated data at all to detect perceived sarcasm in an emerging disaster? If annotated data is vital, can we leverage only a very small set of labeled examples to obtain good performance instead?

Direct Transfer To answer the first question, we stage the following setup: We train our best BERTweet model on out-of-domain sarcasm detection datasets, then *directly* test the model on HURRICANESARC. We experiment with iSarcasm (Oprea and Magdy, 2020) and SARC (Khodak et al., 2017) as our out-of-domain datasets, which both cover a general domain (Twitter and Reddit, respectively), whereas HURRICANESARC covers a specialized domain (Twitter). Since SARC is much larger in size compared with the other datasets, we downsample SARC to $5,000$ examples (which is roughly the size of iSarcasm) to discount for the impact of dataset size. We show the results of these experiments in Table 10 and make a few observations. First, training on SARC or iSarcasm and testing on HURRICANESARC hurts the performance considerably, decreasing the F1 by 18.9% and 15%, respectively, compared with training and testing on HURRICANESARC. It is interesting to see that the drop in performance is much higher when training on Reddit compared with Twitter, which suggests that differences exist in the way sarcasm is expressed across platforms. On the other hand, training on iSarcasm and testing on SARC shows a decrease of only 9% in F1, whereas we note a decrease of 21% in F1 when training on HURRICANESARC and testing on SARC. These results reinforce that in-domain labeled data is vital in providing good model performance and understanding

METHOD	P	R	F1
CARER	0.683	0.705	0.693
EMONET	0.679	0.783	0.781
EMONET SENT	0.651	0.653	0.651
ISARCASM	0.623	0.788	0.701
GOEMOTIONS	0.612	0.622	0.617
SARC	0.671	0.713	0.683
EMONET	0.651	0.681	0.673
SARC	0.680	0.701	0.694
HURRICANE-EXT	0.691	0.718	0.699

Table 12: Precision, recall, and F1 using the following methods: **1)** Supervised Intermediate Pre-training using BERT (upper block). **2)** Unsupervised Pre-training using BERT (lower block).

of the ongoing disaster. However, given the rapid unfolding of a disaster and considering that data annotation is a time-intensive process, only a small amount of annotations can be obtained as the disaster starts to unfold. Therefore, we now turn to our second question, and propose semi-supervised learning approaches as the solution.

Semi-supervised learning Our proposed methods can use large amounts of unlabeled data generated during natural disasters to considerably reduce the annotation costs as well as reduce the time needed to acquire the labeled data. We explore two state-of-the-art semi-supervised techniques, and train these methods on small subsets of our training set. Using a quarter of the entire training set, these methods perform similarly with techniques leveraging the whole set of labeled examples. We consider the following methods: **1) Noisy student training** (Xie et al., 2020) is a approach leveraging knowledge distillation and self-training, which

METHOD	P	R	F1	STDEV
BERT-100	0.583	0.575	0.584	±2.9
BERT-200	0.632	0.623	0.625	±2.1
BERT-ALL	0.663	0.707	0.688	±0.98
NOISY STUDENT-100	0.593	0.595	0.598	±3.1
NOISY STUDENT-200	0.662	0.665	0.663	±2.0
NOISY STUDENT-ALL	0.732	0.734	0.733†	±1.5
UST-100	0.623	0.602	0.616	±2.7
UST-200	0.683	0.666	0.674†	±1.9
UST-ALL	0.752	0.745	0.743†	±0.78

Table 13: Average precision, recall, F-1 and F-1 standard deviation (of the 5 runs) using semi-supervised learning and the vanilla BERT model. The number following the model name indicates the number of examples that model was trained on (e.g., BERT-30 is the vanilla BERT model trained on 30 labeled examples). We assert significance† if $p < 0.05$ under a paired-t test with the counterpart vanilla BERT model (e.g., UST-200 vs. BERT-200).

iteratively jointly trains two models in a teacher-student framework. Noisy student uses a larger model size and noised inputs, exposing the student to more difficult learning environments, which usually leads to an increased performance compared to the teacher. To add noise to our input examples, we use two approaches: a) *Synonym replacement:* We replace between one and three words in a tweet with its synonym using the WordNet English lexical database (Fellbaum, 2012); b) *Back-translation:* We use back-translation, and experiment with different levels of noise corresponding to different translation chain lengths (e.g., English-French-Spanish-English). Smaller chain lengths lead to less noise, while increasing the length of the chain produces examples with significantly more noise. **2) Uncertainty aware Self-Training** (Mukherjee and Awadallah, 2020) incorporates uncertainty estimates into the standard *teacher-student* self-training framework by adding a few highly effective changes to the typical teacher-student self-training framework, such as acquisition functions using Monte Carlo dropout and a new learning mechanism leveraging teacher model confidence.

Few-shot Large Language Models Large language models such as ChatGPT have shown impressive performance in predictive tasks with little to no requirements for labeled data. Therefore despite their large computational costs, we argue that such models can be used in time-critical situations. We leverage the Chat Completions API [1] provided by OpenAI to benchmark ChatGPT. We leverage the following instruction for the model: *Your task is to identify if the author of the following [tweet] is sarcastic about the discussed topic or expresses irony. Please answer with 'yes' or 'no' Here is the [tweet]*:. The ChatGPT version used is 'gpt-3.5-turbo-0125'. We experimented with both zero-shot and few-shot. In few-shot, we randomly appended to the instructions 5 tweets containing sarcasm and 5 tweets that do not express sarcasm.

Reducing Time & Annotation Costs We investigate our semi-supervised learning methods and LLMs can help reduce the human effort needed to annotate the data. To this end, we train the SSL models on subsets of our dataset with only 100 and with only 200 training examples per class. These 100- and 200-size subsets are created randomly. We run each of these methods with 10 different randomly chosen subsets, and report the mean and standard deviation of the obtained F1 for each model. The few-shot ChatGPT only uses 10 input tweets by means of in-context learning.

Results We show the results of the mentioned approaches in Table 11. We make the following observations. Our experiments involving 200 (UST-200) training examples per class show the feasibility of using semi-supervised approaches for sarcasm detection. Using as little as 25% of the available data, semi-supervised approaches perform similar to models trained on the whole dataset. Therefore, the semi-supervised approaches can deliver the same performance but use four times less annotated data. Since disasters are time-critical events, these results show that SSL methods can significantly contribute to a faster and more reliable understanding of an unfolding disaster. We observe that ChatGPT still lags behind our UST-100 (i.e., trained with 100 examples) model significantly. Specifically, ChatGPT obtains an F1 of 0.603 while our approach yields 0.631. Moreover, the F1 variance of different runs of all the UST (UST-100, UST-200, and UST-ALL) methods is significantly improved compared

[1] https://platform.openai.com/docs/guides/text-generation/chat-completions-api

to the vanilla BERTweet, which shows that semi-supervised models are considerably more stable.

BERT Results For completeness, we also show results of the intermediate-task transfer learning and semi-supervised learning experiments using the plain BERT (Devlin et al., 2019) model in Table 12 and Table 13 where we observe similar performance trends.

7. Conclusion

In this paper, we introduced HurricaneSARC, a dataset for perceived sarcasm detection, composed of $15,000$ English tweets from multiple hurricane events, and annotated with the *sarcastic* and *non-sarcastic* labels. We detailed the potential particularities and challenges of our dataset, and developed neural baselines for HurricaneSARC. Next, we improved the performance of our pre-trained language models using supervised or unsupervised intermediate task transfer learning, as well as semi-supervised learning techniques. The best BERTweet model (UST-ALL) is able to improve the performance of our baselines by 5.3% F-1 score, by leveraging the large amounts of unlabeled data from the disaster domain. We hope that our dataset for sarcasm detection in a specific domain will spur research in this area and will lead to novel approaches for in-domain and out-of-domain explorations that will improve natural language understanding of disaster-related tweets as well as the overall understanding of sarcasm. Furthermore, the construction of other specialized datasets, e.g., from a financial domain, is another interesting direction.

Acknowledgements

This research is partially supported by National Science Foundation (NSF) grants IIS-2107524 and IIS-2107487. We also thank our reviewers for their insightful comments.

8. Bibliographical References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. EmoNet: Fine-Grained Emotion Detection with Gated Recurrent Neural Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 718–728, Vancouver, Canada. Association for Computational Linguistics.

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*.

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

Paula Carvalho, Luís Sarmento, Mário J Silva, and Eugénio De Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

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

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054. 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.

Christiane Fellbaum. 2012. Wordnet. *The encyclopedia of applied linguistics*.

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.

Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: a closer look. In *Proceedings of the*

49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 581–586.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *ArXiv*, abs/2004.10964.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, Hong Kong, China. Association for Computational Linguistics.

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

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

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *ACL*.

Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. 2016. Are word embedding-based features useful for sarcasm detection? *arXiv preprint arXiv:1610.00883*.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A large self-annotated corpus for sarcasm. *CoRR*, abs/1704.05579.

Yoon Kim. 2014a. Convolutional neural networks for sentence classification. In *EMNLP*.

Yoon Kim. 2014b. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Roger Kreuz and Gina Caucci. 2007. Lexical influences on the perception of sarcasm. In *Proceedings of the Workshop on computational approaches to Figurative Language*, pages 1–4.

CC Liebrecht, FA Kunneman, and APJ van Den Bosch. 2013. The perfect solution for detecting sarcasm in tweets# not.

Stephanie Lukin and Marilyn Walker. 2013. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 30–40, Atlanta, Georgia.

Navonil Majumder, Soujanya Poria, Haiyun Peng, Niyati Chhaya, Erik Cambria, and Alexander F. Gelbukh. 2019. Sentiment and sarcasm classification with multitask learning. *CoRR*, abs/1901.08014.

Diana Maynard and Mark Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4238–4243, Reykjavik, Iceland. European Language Resources Association (ELRA).

Abhijit Mishra, Diptesh Kanojia, and Pushpak Bhattacharyya. 2016. Predicting readers' sarcasm understandability by modeling gaze behavior. In *AAAI*.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. Uncertainty-aware self-training for text classification with few labels. *arXiv preprint arXiv:2006.15315*.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Silviu Oprea and Walid Magdy. 2020. isarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–41, Los Angeles. Association for Computational Linguistics.

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.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier.

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

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223.

Jishnu Ray Chowdhury, Cornelia Caragea, and Doina Caragea. 2019. Keyphrase Extraction from Disaster-related Tweets. In *The World Wide Web Conference*, WWW 2019, pages 1555–1566, New York, NY, USA. Association for Computing Machinery (ACM).

Antonio Reyes and Paolo Rosso. 2012. Making objective decisions from subjective data: Detecting irony in customer reviews. *Decision support systems*, 53(4):754–760.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language resources and evaluation*, 47(1):239–268.

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.

Hannah Ritchie and Max Roser. 2020. Natural Disasters. *Our World in Data*.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium.

Edoardo Savini and Cornelia Caragea. 2020. A multi-task learning approach to sarcasm detection (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(10):13907–13908.

Edoardo Savini and Cornelia Caragea. 2022. Intermediate-task transfer learning with bert for sarcasm detection. *Mathematics*, 10(5).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

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

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

Tony Veale and Yanfen Hao. 2010. Detecting ironic intent in creative comparisons. In *ECAI*, volume 215, pages 765–770.

Byron C Wallace, Eugene Charniak, et al. 2015. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1035–1044.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698.

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.