Decay No More: A Persistent Twitter Dataset for Learning Social Meaning

Chiyu Zhang Muhammad Abdul-Mageed El Moatez Billah Nagoudi
Deep Learning & Natural Language Processing Group
The University of British Columbia
chiyuzh@mail.ubc.ca, {muhammad.mageed,moatez.nagoudi}@ubc.ca

Abstract

With the proliferation of social media, many studies resort to social media to construct datasets for developing social meaning understanding systems. For the popular case of Twitter, most researchers distribute tweet IDs without the actual text contents due to the data distribution policy of the platform. One issue is that the posts become increasingly inaccessible over time, which leads to unfair comparisons and a temporal bias in social media research. To alleviate this challenge of data decay, we leverage a paraphrase model to propose a new *persistent* English Twitter dataset for social meaning (PTSM). PTSM consists of 17 social meaning datasets in 10 categories of tasks. We experiment with two SOTA pre-trained language models and show that our PTSM can substitute the actual tweets with paraphrases with *marginal* performance loss.[1]

Original: **(Task: emotion, Label: joy)**

- Being stuck in the roof of your house provides amazing view 😍 but sheer terror of falling down, kinda like life.

Paraphrase:

- Living on a roof from a roof creates incredible views yet the sheer terror of falling down fits within.
- Beyond and up! Finding yourself in the roof gives you amazing views but sheer terror of falling...
- From the roof of your house, you have a great view but you're scared of falling down.

Figure 1: Paraphrase example from $\text{Emo}_{\text{Moham}}$.

1 Introduction

Social meaning is substantiated in socio-pragmatics, and refers to intended meaning in real-world communication (Thomas 2014) and how utterances should be interpreted within the social context in which they are produced. Aspects of social meaning include emotion recognition (Mohammad et al. 2018), irony detection (Van Hee, Lefever, and Hoste 2018), sarcasm detection (Riloff et al. 2013; Bamman and Smith 2015), hate speech identification (Waseem and Hovy 2016), and stance identification (Mohammad et al. 2016). A successful social meaning comprehension system can ameliorate a wide range of NLP applications. For example, a dialogue system with knowledge of social meaning can provide more engaging reactions when it realizes human emotions.

With the proliferation of social media, billions of users share content in various forms. Social media platforms, such as Twitter, allow users to express their opinions, discuss topics, and connect with friends, among other practices (Farzindar and Inkpen 2015). As a result, social media offers abundant resources for social meaning understanding. Over the years, researchers have developed a number of labelled datasets to train (semi-)supervised machine learning models. According to the data redistribution policy of a social media platform such as Twitter, a post's actual content cannot be shared with third parties. Hence, many studies only distribute the IDs of posts (e.g., tweets). The challenge with this set up, however, substantial social media posts become inaccessible over time due to deletion, protection, etc. (Zubiaga 2018; Assenmacher et al. 2021). We empirically characterize this issue of data inaccessibility by attempting to re-collect the tweet contents of six social meaning datasets. As Table 1 shows, we could only acquire, on average, 73% of the tweets. This data decay leads to a serious issue concerning the lack of replicability in social media studies. The issue of data inaccessibility also introduce a temporal bias (as older data sets become significantly unavailable) and leads to unfair comparisons. In other words, it is difficult to compare models trained on differently sized datasets. In this work, we propose a *persistent* Twitter dataset for social meaning (PTSM) to alleviate these issues. We use a state-of-the-art (SOTA) Transformer-based pre-trained model, i.e., T5 (Raffel et al. 2020), as backbone to train a paraphrasing model with diverse parallel data. We then use the trained model to produce sentential paraphrases of training samples. We then benchmark our PTSM with two SOTA pre-trained language models (PLM), RoBERTa (Liu et al. 2019) and BERTweet (Nguyen, Vu, and Tuan Nguyen 2020). Our experiment results show that we can replace the actual tweets with their paraphrases without sacrificing performance: On average, our paraphrase-based models are only $\sim$ 1.70 F_1 below models trained with original data across the 17 datasets.

To summarize, we make the following **contributions**: **(1)**

[1]Our data is available at: https://github.com/chiyuzhang94/PTSM.

We introduce PTSM, a persistent Twitter dataset for social meaning comprising 17 different datasets, whose accessibility is enhanced via paraphrasing. To the best of our knowledge, this is the first work to employ SOTA paraphrase methods to tackle occasional data decay for social media research. **(2)** We develop a Transformer-based model for paraphrasing social media data. **(3)** We benchmark our PTSM with two SOTA PLMs and demonstrate the promise of training tweet classifiers with paraphrases only.

	Prod. year	Orig.	Retr.	Decay %
Riloff et al. (2013)	2013	3.0K	1.8K	0.41
Ptáček, Habernal, and Hong (2014)	2014	100.0K	89.3K	0.11
Rajadesingan, Zafarani, and Liu (2015)	2015	91.0K	51.6K	0.43
Bamman and Smith (2015)	2015	19.5K	14.8K	0.34
Waseem and Hovy (2016)	2016	16.9K	10.9K	0.36
Rosenthal, Farra, and Nakov (2017)	2017	50.3K	48.2K	0.04

Table 1: Issue of data inaccessibility. These six datasets were distributed by their authors via tweet IDs. **Orig.:** original size of each dataset. **Retr.:** data we successfully collected via Twitter API in November, 2020. **Decay %:** percentage of inaccessible tweets.

2 Related Work

Paraphrasing aims at rewriting or rephrasing a text while maintaining its original semantics. Most previous works introduce paraphrasing as a way to augment training data and hence alleviate data sparsity in machine learning models. To reduce the high degree of lexical variation, Petrović, Osborne, and Lavrenko (2012); Li et al. (2016) produce paraphrases of tweets by replacing words of original text with their synonyms based on WordNet and word embedding vector closeness. Beddiar, Jahan, and Oussalah (2021) introduce a back-translation method to augment training data for hate speech detection. Different to the previous works, we utilize paraphrases to develop a tweet classifier without any subsequent use of the original tweets while training our downstream models. Our main objective is to tackle the data decay issue in machine learning of social media. For that, we offer a unified paraphrase dataset for training social meaning models that is directly comparable to original training data for a host of tasks. Our work has affinity to work aiming at facilitating meaningful model comparisons such as the general language understanding evaluation (GLUE) benchmark (Wang et al. 2019) and SuperGLUE (Sarlin et al. 2020), but we focus on availing training data that enable model building in the first place. Barbieri et al. (2020) introduce TweetEval, a benchmark for tweet classification evaluation, but they are not able to share more than 50K tweets per dataset due to Twitter distribution policies. Different to them, we are able to provide unlimited numbers of paraphrases for future research.

3 Persistent Dataset for Social Meaning

3.1 Social Meaning Datasets

We collect 17 datasets representing 10 different social meaning tasks, as follows: [2]

[2]To facilitate reference, we give each dataset a name.

Task	Classes	Train	Dev	Test	Total
$\text{Crisis}_{\text{Oltea}}$	{on-topic, off-topic,}	48.0K	6.0K	6.0K	60.0K
$\text{Emo}_{\text{Moham}}$	{anger, joy, opt., sad.}	3.3K	374	1.4K	5.0K
Hate_{Bas}	{hateful, none}	9.0K	1.0	3.0	13.0K
$\text{Hate}_{\text{Waseem}}$	{racism, sexism, none}	8.7K	1.1K	1.1K	10.9K
$\text{Hate}_{\text{David}}$	{hate, offensive, neither}	19.8K	2.5K	2.5K	24.8K
$\text{Humor}_{\text{Potash}}$	{humor, not humor}	11.3K	660	749	12.7K
$\text{Humor}_{\text{Meaney}}$	{humor, not humor}	8.0K	1.0K	1.0K	10.0K
$\text{Irony}_{\text{Hee-A}}$	{ironic, not ironic}	3.5K	384	784	4.6K
$\text{Irony}_{\text{Hee-B}}$	{IC, SI, OI, NI}	3.5K	384	784	4.6K
$\text{Offense}_{\text{Zamp}}$	{offensive, not offensive}	11.9K	1.3K	860	14.1K
$\text{Sarc}_{\text{Riloff}}$	{sarcastic, non-sarcastic}	1.4K	177	177	1.8K
$\text{Sarc}_{\text{Ptacek}}$	{sarcastic, non-sarcastic}	71.4K	8.9K	8.9K	89.3K
$\text{Sarc}_{\text{Rajad}}$	{sarcastic, non-sarcastic}	41.3K	5.2K	5.2K	51.6K
Sarc_{Bam}	{sarcastic, non-sarcastic}	11.9K	1.5K	1.5K	14.8K
$\text{Senti}_{\text{Rosen}}$	{neg., neu., pos.}	42.8K	4.8K	12.3K	59.8K
$\text{Senti}_{\text{Thel}}$	{neg., pos.}	900	100	1.1K	2.1K
$\text{Stance}_{\text{Moham}}$	{against, favor, none}	2.6K	292	1.3K	4.2K

Table 2: Gold social meaning datasets. **opt.:**: Optimism, **sad.:** Sadness, **IC:** Ironic by clash, **SI:** Situational irony, **OI:** Other irony, **NI:** Non-ironic, **neg.:** Negative, **Neu.:** Neutral, **pos.:** Positive.

Crisis awareness. We use $\text{Crisis}_{\text{Oltea}}$ (Olteanu et al. 2014), a corpus for identifying whether a tweet is related to a given disaster or not.

Emotion. We utilize $\text{Emo}_{\text{Moham}}$, introduced by Mohammad et al. (2018), for emotion recognition. We use the version adapted in Barbieri et al. (2020).

Hateful and offensive language. We use Hate_{Bas} (Basile et al. 2019), $\text{Hate}_{\text{Waseem}}$ (Waseem and Hovy 2016), $\text{Hate}_{\text{David}}$ (Davidson et al. 2017), and $\text{Offense}_{\text{Zamp}}$ (Zampieri et al. 2019).

Humor. We use the humor detection datasets $\text{Humor}_{\text{Potash}}$ (Potash, Romanov, and Rumshisky 2017) and $\text{Humor}_{\text{Meaney}}$ (Meaney et al. 2021).

Irony. We utilize $\text{Irony}_{\text{Hee-A}}$ (irony detection) and $\text{Irony}_{\text{Hee-B}}$ (irony type identification) from Van Hee, Lefever, and Hoste (2018).

Sarcasm. We use four sarcasm datasets from $\text{Sarc}_{\text{Riloff}}$ (Riloff et al. 2013), $\text{Sarc}_{\text{Ptacek}}$ (Ptáček, Habernal, and Hong 2014), $\text{Sarc}_{\text{Rajad}}$ (Rajadesingan, Zafarani, and Liu 2015), and Sarc_{Bam} (Bamman and Smith 2015).

Sentiment. We employ the three-way sentiment analysis dataset from $\text{Senti}_{\text{Rosen}}$ (Rosenthal, Farra, and Nakov 2017) and a binary sentiment analysis dataset from $\text{Senti}_{\text{Thel}}$ (Thelwall, Buckley, and Paltoglou 2012).

Stance. We use $\text{Stance}_{\text{Moham}}$, a stance detection dataset from Mohammad et al. (2016). The task is to identify the position of a given tweet towards a target of interest.

Data Crawling and Preparation. We use the Twitter API [3] to crawl datasets which are available only in tweet ID form. Before we paraphrase the data or use it in our various experiments, we normalize each tweet by replacing the user names and hyperlinks to the special tokens 'USER' and 'URL', respectively. This ensures our paraphrased dataset will never have any actual usernames or hyperlinks, thereby protecting user identity. For datasets collected based on hashtags by original authors, we also remove the seed hashtags from the original tweets. For datasets originally used in cross-

[3]https://developer.twitter.com/

validation, we acquire 80% Train, 10% Dev, and 10% Test via random splits. For datasets that had training and test splits but not development data, we split off 10% from training data into Dev. The splits of each dataset are presented in Table 2.

3.2 Praphrase Model

In order to train our paraphrasing model, we collect four paraphrase datasets from PIT-2015 (Xu, Callison-Burch, and Dolan 2015), LanguageNet (Lan et al. 2017), Opusparcus (Creutz 2018), and Quora Question Pairs (QQP) (Iyer et al. 2017). We only keep sentence pairs with a high semantic similarity as follows: (1) For PIT-2015, we extract sentence pairs with semantic similarity labels of 5 and 4. (2) For LanguageNet, we keep sentence pairs obtained with similarity labels of 4, 5, and 6. (3) For Opusparcus, we take pairs whose similarity labels are 4. (4) For QQP, we extract sentence pairs with the 'duplicate' label. Table 3 presents the data size of each corpus after filtering. We then merge all extracted samples (a total of $625,097$ pairs) and split them into Train, Dev, and Test (80%, 10%, and 10%).

Dataset	Domain	# of samples
PIT-2015	Tweet	3,789
LanguageNet	Tweet	12,988
Opusparcus	Video subtitle	462,846
QQP	Quora	149,263
Total		625,097

Table 3: Paraphrasing datasets.

For the modelling, we fine-tune a pre-trained T5_{Base} (Raffel et al. 2020) on the Train split for 20 epochs with a constant learning rate of $3e-4$ and a maximal sequence length of 512. We evaluated the model on the Dev set at the end of each epoch and identified the best model (28.18 BLEU score).

3.3 Generating PTSM

We apply the trained paraphrasing model on the Train split of each of our 17 social meaning datasets, using top-p sampling (Holtzman et al. 2020) with $p = 0.95$ to generate 10 paraphrases for each gold sample. We then select paraphrases of a given tweet based on the following criteria: **(i)** To remove any 'paraphrases' that are just copies or near-duplicates of the original tweet, we use a simple tri-gram similarity method where we exclude any paraphrases whose similarity with the original tweet is > 0.95. **(ii)** To ensure that paraphrases are not totally different from the original tweets, we remove any generations whose tri-gram similarity with the original tweet $= 0$. **(iii)** We then sort the remaining paraphrases by their tri-gram similarities with the original tweet and descendingly add paraphrase into set P. To populate P, we proceed as follows: we loop over paraphrases one by one, calculating the similarity of each each other paraphrase in P. If the paraphrase is > 0.50 similar to any item in P, we do not it add into P. Ultimately, each original tweet associates with a set of paraphrases P. This process results in a paraphrasing dataset `Para-Train-Clean`. We present example paraphrases in Figure 1 and Table 5. We observe that the paraphrase model fails to generate emojis since emojis are out-of-vocabulary for the original T5 model, but the model can preserve the overall semantics of the original tweet. To explore effect of the size of paraphrase data on the downstream tasks, we extract 1, 2, 4, and 5 paraphrases from `Para-Train-Clean` for each Train gold sample in each of our 17 tasks. We refer to the resulting datasets as `Para1`, `Para2`, `Para4`, and `Para5`. Although the main puprpose of this paper is to solve the data decay issue of training data, we also provide a paraphrase dataset for Dev and Test splits of each of our 17 social meaning datasets. For each Dev or Test gold sample, we extract one paraphrase sample. We refer to the resulting datasets as `Para-Dev` and `Para-Test`. Table 4 shows the distribution of the resulting paraphrase datasets.[4]

Task	Para1	Para2	Para4	Para5	ParaD	ParaT
$\text{Crisis}_{\text{Oltea}}$	48.0K	86.8K	120.7K	123.9K	6.0K	6.0K
$\text{Emo}_{\text{Moham}}$	3.3K	6.3K	10.9K	12.2K	374	1.4K
Hate_{Bas}	9.0K	17.6K	31.1K	35.1K	1.0	3.0
$\text{Hate}_{\text{Waseem}}$	8.7K	16.6K	28.3K	31.7K	1.1K	1.1K
$\text{Hate}_{\text{David}}$	19.8K	38.2K	65.5K	73.4K	2.5K	2.5K
$\text{Humor}_{\text{Potash}}$	11.3K	21.8K	38.3K	44.0K	660	749
$\text{Humor}_{\text{Meaney}}$	8.0K	15.7K	28.7K	33.0K	1.0K	1.0K
$\text{Irony}_{\text{Hee-A}}$	3.5K	6.6K	11.4K	12.8K	384	784
$\text{Irony}_{\text{Hee-B}}$	3.5K	6.6K	11.5K	12.9K	384	784
$\text{Offense}_{\text{Zamp}}$	11.9K	23.0K	39.6K	44.3K	1.3K	860
$\text{Sarc}_{\text{Riloff}}$	1.4K	2.7K	4.6K	5.2K	177	177
$\text{Sarc}_{\text{Ptacek}}$	71.4K	138.9K	242.1K	272.4K	8.9K	8.9K
$\text{Sarc}_{\text{Rajad}}$	41.3K	78.3K	131.5K	146.6K	5.2K	5.2K
Sarc_{Bam}	11.9K	22.4K	37.5K	41.6K	1.5K	1.5K
$\text{Senti}_{\text{Rosen}}$	42.8K	84.3K	154.8K	178.1K	4.8K	12.3K
$\text{Senti}_{\text{Thel}}$	900	1.7K	3.1K	3.5K	100	1.1K
$\text{Stance}_{\text{Moham}}$	2.6K	4.7K	6.4K	6.6K	292	1.3K

Table 4: Distribution of PTSM. **Para**n indicates that the Train set with varying paraphrase size. **ParaD:** Para-Dev, **ParaT:** Para-Test.

4 Experiment and Result

4.1 Implementation

We evaluate the quality of the 17 paraphrased Train datasets in Table 4 via fine-tuning two Transformer-based pre-trained language models, i.e, $\text{RoBERTa}_{\text{Base}}$ (Liu et al. 2019) and $\text{BERTweet}_{\text{Base}}$ (Nguyen, Vu, and Tuan Nguyen 2020). We utilize the checkpoints released by Huggingface[5]. For $\text{Crisis}_{\text{Oltea}}$ and $\text{Stance}_{\text{Moham}}$, we append the topic term behind the post content, separate them by an '[SEP]' token, and set maximal sequence length to 72. For the rest of tasks, we set maximal sequence length to 64. For all the tasks, we pass the hidden state of '[CLS]' token from the last Transformer encoder layer through two feed-forward layers (a linear layer with a $tanh$ activation function followed by another linear layer with a $softmax$ activation function) to predict the classification label. Cross-Entropy is used to calculate the training loss. Following Nguyen, Vu, and Tuan Nguyen (2020), we use Adam with a peak learning rate of $1e-5$ and a weight decay of 0.01 to optimize the model and fine-tune each task for 20 epochs with early stopping ($patience = 5$ epochs). The training batch size is 32. We compare the model trained

[4]Note that some samples may not have enough paraphrases to construct Para2, Para4, and Para5.

[5]`https://huggingface.co/models`

Original Tweet	Paraphrase	Label
USER but what I am doing is in my control, #AvoidMMT , you guys are #terrible	You guys are horrible, avoid MMT what I am doing is in my control, #AvoidMMT, you guys are terrifying You guys are #terrorist. I have used everything I have to do. You guys are awful, but I am going to stop doing it.	anger
This is the 47th wedding I've ruined. When I hit 50 I can retire.	The 46th wedding I've ruined. When I hit 50 I can retire. It's nice to see yo Here's the 47th wedding I've ruined. If I'm old enough to go on the 40s I can get married. After a single wedding, I drew 47 weddings, and before I hit 50 I can retire"	humor
Sorry guys I have absolutely no idea what time i'll be on cam tomorrow but will keep you posted.	Sorry to disturb you. I have absolutely no idea what time I'll be on cam tomorrow. I have absolutely no idea what time I'll be on camera tomorrow but I'll keep you posted	sadness
Who wants to go with me to get my tattoo tomorrow? I'll buy you Dunkin doughnuts	I'll buy you Dunkin' Donuts for $5. Who wants to go with me for my tattoo tomorrow? I'll buy you a Dunkin' Donuts. Who wants to go with me to get my tattoo tomorrow?	neutral
USER May I suggest, that you have a meal that is made with beans, onions & garlic, the day before class.	The day before class please eat beans, onions and garlic. Also see the videos The Day Before Class. You should make that meal, (do you think). If you can eat just the day before class, make a wonderful meal with garlic, onions and beans.	joy

Table 5: More examples of paraphrases in PTSM.

Task	RoBERTa									BERTweet								
	Gold	P1	P2	P4	P5	P1+G	P2+G	P4+G	P5+G	Gold	P1	P2	P4	P5	P1+G	P2+G	P4+G	P5+G
$\text{Crisis}_{\text{O.}}$	95.96	**95.38**	95.09	95.18	95.00	**95.88**	95.85	95.55	95.17	95.58	94.90	95.11	**95.26**	95.17	**95.75**	95.60	95.47	95.63
$\text{Emo}_{\text{M.}}$	77.61	76.29	**77.16**	77.02	76.59	78.54	77.65	**78.82**	78.54	80.37	77.61	78.06	77.66	**78.62**	**80.71**	80.29	78.91	79.56
$\text{Hate}_{\text{B.}}$	49.74	44.70	46.48	45.81	**47.89**	**48.78**	47.99	48.26	48.21	56.51	**54.01**	52.41	51.71	52.04	**54.88**	54.53	53.01	51.89
$\text{Hate}_{\text{W.}}$	56.84	54.22	**55.00**	54.89	54.92	56.64	**56.67**	56.17	55.46	57.07	55.33	55.41	**59.33**	55.90	56.42	**56.87**	56.10	56.14
$\text{Hate}_{\text{D.}}$	77.03	75.14	73.79	74.43	**75.20**	**75.81**	75.57	74.66	74.66	77.57	75.11	**75.75**	74.35	75.08	**77.21**	74.70	76.00	75.97
$\text{Humor}_{\text{P.}}$	54.66	52.92	**55.81**	54.18	52.59	55.11	**56.10**	53.25	50.64	57.56	**52.77**	52.65	52.11	50.92	**56.13**	53.42	53.77	54.34
$\text{Humor}_{\text{M.}}$	92.61	90.96	**92.25**	91.20	91.96	91.59	92.08	**92.45**	91.57	94.21	93.31	93.30	**93.76**	92.67	93.95	94.00	93.57	**94.16**
$\text{Irony}_{\text{H-A}}$	73.13	69.96	70.46	70.52	**71.14**	72.51	**72.72**	72.22	72.06	76.82	**76.26**	75.42	75.08	75.37	**77.90**	77.43	77.04	75.60
$\text{Irony}_{\text{H-B}}$	51.56	46.23	**48.52**	46.92	45.31	**50.54**	49.20	49.79	47.17	56.84	47.89	51.18	49.98	**51.78**	56.24	**56.45**	50.72	55.63
$\text{Offense}_{\text{Z}}$	80.67	77.49	**80.22**	80.18	79.41	80.39	80.20	**80.77**	79.94	79.74	77.63	78.89	79.29	**80.15**	78.39	78.56	**79.47**	79.32
$\text{Sarc}_{\text{Ri.}}$	75.28	71.04	70.50	71.22	**72.64**	71.44	71.69	73.39	**74.52**	76.61	**79.97**	78.61	78.81	75.97	**80.19**	77.01	77.60	77.21
$\text{Sarc}_{\text{P.}}$	95.59	91.67	93.48	94.03	**94.35**	**95.34**	**95.34**	95.11	95.16	96.74	92.96	94.34	94.62	**94.70**	**96.34**	96.04	95.85	95.71
$\text{Sarc}_{\text{Ra.}}$	85.64	80.82	**82.34**	82.20	82.29	**84.90**	84.71	84.00	84.29	86.97	84.08	84.99	84.81	**85.31**	**86.95**	86.55	86.14	86.19
$\text{Sarc}_{\text{B.}}$	80.01	77.33	**78.18**	77.69	77.09	**79.88**	79.49	78.91	78.56	82.59	80.45	80.60	**80.93**	80.32	**82.92**	82.11	81.99	81.98
$\text{Senti}_{\text{R.}}$	70.78	70.59	**71.44**	70.26	69.99	**71.19**	71.15	70.30	70.76	72.05	**71.23**	70.21	69.75	69.98	71.67	71.19	71.17	**71.78**
$\text{Senti}_{\text{T.}}$	88.99	87.52	**88.12**	87.83	87.79	**89.04**	88.46	88.48	87.62	89.27	88.18	89.00	88.49	**89.28**	89.11	89.35	89.16	**89.64**
$\text{Stance}_{\text{M.}}$	69.28	67.81	68.27	**69.76**	69.07	68.56	**68.64**	68.44	67.54	69.11	67.30	**67.92**	66.96	66.76	69.12	**69.83**	68.11	66.74
Average	75.02	72.36	**73.36**	73.14	73.13	**74.48**	74.33	74.15	73.64	76.80	74.65	**74.93**	74.88	74.71	**76.70**	76.11	75.53	75.73

Table 6: Benchmarking PTSM. **Gold** denotes a model fine-tuned with downstream, original Train data. **P**n indicates that the model is trained on Para$n$ training set. **P**n**+G** indicates that the model is trained on the combination of Para$n$ and original gold training set. **Bold** denotes the best result for each task under each group of settings. **Underscore** indicates that the model outperforms the corresponding baseline that is fine-tuned on gold Train set.

on PTSM to ones fine-tuned on the original gold Train set with the same hyper-parameters. We run three times with random seeds for all downstream fine-tuning, *reporting the average of these three runs*. All downstream task models are fine-tuned on an Nvidia V100 GPUs (32G). For individual task, we typically identify the best model on each respective Dev set and evaluate its performance on blind Test. We present the average Test macro-averaged F_1 over the three runs as mentioned, and introduce a global metric averaging the macro-F_1 scores over the 17 datasets.

4.2 Results

We use our PTSM to investigate the viability of using paraphrased training data instead of gold training sets. We fine-tune PLMs on the PTSM Train sets with varying paraphrase sizes (referred to as Pn in Table 6) but evaluate on the original Dev and Test sets for all of the individual tasks. As Table 6 shows, although none of our paraphrase-based models exceed the corresponding baseline model that is fine-tuned on gold (i.e., original) Train sets in term of average F_1, the paraphrase-based models either slightly exceeds or approaches performance of the gold models on individual datasets. Regarding the effect of paraphrase data size, we find P2 Train to perform best both for RoBERTa and BERTweet models as compared to other amounts of paraphrase data. This shows that while doubling paraphrase data size is useful, more paraphrases do not help the models. RoBERTa-P2 (the best setting of paraphrase-based RoBERTa fine-tuning) obtains an average F_1 of 73.36, which is 1.66 less than RoBERTa-Gold (while outperforming the latter on $\text{Humor}_{\text{Potash-17}}$ and $\text{Senti}_{\text{Rosen-17}}$). BERTweet-P2 (the best setting of paraphrase-based BERTweet fine-tuning) underperforms BERTweet-Gold with 1.87 average F_1 (i.e., 74.93). We also observe our BERTweet-P1 model achieves a sizable improvement of 3.36 F_1 on $\text{Sarc}_{\text{Riloff}}$ over the gold model. These findings demonstrate that **(i)** we can replace social gold data (which can become increasingly *inaccessible* over time) with paraphrase datasets (which are *persistent*) **(ii)** without

sacrificing much performance. In addition, we also fine-tune PLMs on the combination of gold Train and our paraphrase Train sets (referred to as Pn+G in Table 6), but find average F_1 scores of our models to still remain below the models fine-tuned on gold data only. We hypothesize that one limitation of our paraphrased Train sets is the lack of emojis. However, analyzing presence of emoji contents for each of the datasets, we find that our paraphrase-based models are able to acquire comparable performance to models trained with original datasets that do not employ any emojis (e.g., none of the training samples of $\text{Senti}_{\text{Rosen-17}}$, $\text{Senti}_{\text{Thel}}$, and $\text{Stance}_{\text{Moham}}$ uses emojis). To further investigate the issue, we fine-tune PLMs on a version of the gold Train set after removing emojis. Here, we observe a slight degradation of performance: RoBERTa and BERTweet each obtains an average F_1 of 74.70 and 76.49, respectively (see Tabel 7). These findings reflect the effect of emoji symbols on social meaning detection. This suggests we can enhance our paraphrase data by inserting the same emojis as original data. We cast this as future work.

Task	RoBERTa		BERTweet	
	Original	RM emoji	Original	RM emoji
$\text{Crisis}_{\text{Oltea}}$	95.96	95.88	95.58	95.70
$\text{Emo}_{\text{Moham}}$	77.61	77.37	80.37	79.77
Hate_{Bas}	49.74	49.19	56.51	55.03
$\text{Hate}_{\text{Waseem}}$	56.84	56.37	57.07	56.93
$\text{Hate}_{\text{David}}$	77.03	77.19	77.57	77.46
$\text{Humor}_{\text{Potash}}$	54.66	54.60	57.56	54.47
$\text{Humor}_{\text{Meaney}}$	92.61	92.38	94.21	94.63
$\text{Irony}_{\text{Hee-A}}$	73.13	72.59	76.82	76.97
$\text{Irony}_{\text{Hee-B}}$	51.56	52.49	56.84	56.80
$\text{Offense}_{\text{Zamp}}$	80.67	79.38	79.74	79.98
$\text{Sarc}_{\text{Riloff}}$	75.28	72.35	76.61	79.60
$\text{Sarc}_{\text{Ptacek}}$	95.59	95.73	96.74	96.45
$\text{Sarc}_{\text{Rajad}}$	85.64	85.35	86.97	86.93
Sarc_{Bam}	80.01	79.02	82.59	82.38
$\text{Senti}_{\text{Rosen}}$	70.78	70.44	72.05	71.56
$\text{Senti}_{\text{Thel}}$	88.99	89.94	89.27	89.13
$\text{Stance}_{\text{Moham}}$	69.28	69.64	69.11	66.54
Average	75.02	74.70	76.80	76.49

Table 7: Effect of emojis in Train. Original indicates the original gold Train set. RM emoji indicates the gold Train set after removing emojis.

Our experiments show that performance of a model trained on PTSM is on par with that of a model trained on gold data. Further, we investigate a scenario where we do not have access to any gold data for development nor test sets. In other words, we evaluate model performance on paraphrased Dev and Test sets under both gold and paraphrased training settings. We do so by fine-tuning RoBERTa on gold and Pn Train sets, using Para-Dev to identify the best model and testing on blind Para-Test. Regardless of the source of training data (gold or paraphrased), we find that models incur a significant performance drop as Table 8 shows. That is, models evaluated on Para-Test drop 7.98 F_1 points (gold-trained model) and 4.84 F_1 points (best paraphrase-trained models) as compared to their respective counterparts evaluated on gold Test. We note that the scenario of no gold test data is not realistic, since it is usually fine to release gold test dataset (much less than 50K data points in most cases).

Task	Gold	P1	P2	P4	P5
$\text{Crisis}_{\text{Oltea}}$	89.67	91.87	92.04	92.11	91.92
$\text{Emo}_{\text{Moham}}$	63.40	63.64	65.25	65.35	64.36
Hate_{Bas}	60.83	54.19	56.85	55.82	55.85
$\text{Hate}_{\text{Waseem}}$	48.90	52.62	52.75	52.49	53.14
$\text{Hate}_{\text{David}}$	58.43	66.40	67.63	66.07	67.72
$\text{Humor}_{\text{Potash}}$	54.34	53.39	52.42	49.58	49.94
$\text{Humor}_{\text{Meaney}}$	82.89	83.92	83.78	83.17	82.82
$\text{Irony}_{\text{Hee-A}}$	63.89	66.07	66.27	67.41	66.19
$\text{Irony}_{\text{Hee-B}}$	40.97	39.09	42.43	42.02	41.74
$\text{Offense}_{\text{Zamp}}$	73.95	72.63	73.08	74.18	74.18
$\text{Sarc}_{\text{Riloff}}$	64.37	66.94	66.17	68.38	67.88
$\text{Sarc}_{\text{Ptacek}}$	83.11	85.80	88.00	89.20	89.54
$\text{Sarc}_{\text{Rajad}}$	73.46	74.85	74.44	75.21	74.87
Sarc_{Bam}	71.03	72.38	73.17	72.91	73.52
$\text{Senti}_{\text{Rosen}}$	64.24	63.62	64.29	64.47	65.28
$\text{Senti}_{\text{Thel}}$	84.10	83.37	83.64	84.85	83.22
$\text{Stance}_{\text{Moham}}$	62.08	62.74	62.56	63.01	62.69
Average	67.04	67.85	68.52	68.60	68.52

Table 8: Testing on Para-Test. **Gold** denotes a model fine-tuned with downstream, original Train data. **P**n indicates that the model is trained on Para$n$ training set.

5 Conclusion and Limitations

Motivated by the issue of decay of social media data, we proposed simple paraphrasing as employed in our PTSM data as a way to avail training datasets for learning social meaning. We fine-tune a T5 model on diverse paraphrasing dataset and utilize the trained model to generate paraphrases of original social meaning Train sets. Through experimental results, we show that we can substitute the actual tweets with their paraphrases while incurring a marginal performance loss. Due to the closed vocabulary of T5, our paraphrasing model cannot generate emojis (even though these can be useful for social meaning detection). We can rectify this by simple post-processing in future work. Another possible limitation of our work is that we use a simple tri-gram similarity method to measure similarity between an actual tweet and its paraphrases. A more sophisticated method may enhance our resulting paraphrase data and hence possibly improve downstream model performance; we will explore this in the future.

Ethical Considerations

PTSM is collected from publicly available sources and aims at availing resources for training NLP models without need for original user data, which could be a step forward toward protecting privacy. Following Twitter policy, all the data we used for model training are anonymous. We also notice the annotation bias existing in the original datasets (e.g., $\text{Hate}_{\text{Waseem}}$).

Acknowledgements

We gratefully acknowledge support from the Natural Sciences and Engineering Research Council of Canada (NSERC; RGPIN-2018-04267), the Social Sciences and Humanities Research Council of Canada (SSHRC; 435-2018-0576; 895-2021-1008), Compute Canada, and UBC ARC-Sockeye.[6]

[6] https://arc.ubc.ca/ubc-arc-sockeye

References

Assenmacher, D.; Weber, D.; Preuss, M.; Calero Valdez, A.; Bradshaw, A.; Ross, B.; Cresci, S.; Trautmann, H.; Neumann, F.; and Grimme, C. 2021. Benchmarking crisis in social media analytics: a solution for the data-sharing problem. *Social Science Computer Review* .

Bamman, D.; and Smith, N. A. 2015. Contextualized sarcasm detection on twitter. In *Proc. of AAAI*.

Barbieri, F.; Camacho-Collados, J.; Espinosa Anke, L.; and Neves, L. 2020. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Findings of the ACL: EMNLP 2020*.

Basile, V.; Bosco, C.; Fersini, E.; Nozza, D.; Patti, V.; Rangel Pardo, F. M.; Rosso, P.; and Sanguinetti, M. 2019. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In *Proc. of SemEval*.

Beddiar, D. R.; Jahan, M. S.; and Oussalah, M. 2021. Data expansion using back translation and paraphrasing for hate speech detection. *Online Social Networks and Media* .

Creutz, M. 2018. Open Subtitles Paraphrase Corpus for Six Languages. In *Proc. of LREC*.

Davidson, T.; Warmsley, D.; Macy, M.; and Weber, I. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proc. of AAAI*.

Farzindar, A.; and Inkpen, D. 2015. Natural language processing for social media. *Synthesis Lectures on Human Language Technologies* .

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2020. The Curious Case of Neural Text Degeneration. In *Proc. of ICLR*.

Iyer, S.; Dandekar, N.; Csernai, K.; et al. 2017. First quora dataset release: Question pairs. *data. quora. com* .

Lan, W.; Qiu, S.; He, H.; and Xu, W. 2017. A Continuously Growing Dataset of Sentential Paraphrases. In *Proceedings of the 2017 Conference on EMNLP*.

Li, Q.; Shah, S.; Ghassemi, M.; Fang, R.; Nourbakhsh, A.; and Liu, X. 2016. Using paraphrases to improve tweet classification: Comparing wordnet and word embedding approaches. In *2016 IEEE International Conference on Big Data*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* .

Meaney, J.; Wilson, S. R.; Chiruzzo, L.; and Magdy, W. 2021. HaHackathon: Detecting and Rating Humor and Offense. In *Proc. of the 59th Annual Meeting of the ACL and the 11th IJCNLP*.

Mohammad, S.; Bravo-Marquez, F.; Salameh, M.; and Kiritchenko, S. 2018. SemEval-2018 Task 1: Affect in Tweets. In *Proc. of SemEval*.

Mohammad, S.; Kiritchenko, S.; Sobhani, P.; Zhu, X.; and Cherry, C. 2016. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proc. of SemEval*.

Nguyen, D. Q.; Vu, T.; and Tuan Nguyen, A. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proc. of the 2020 Conference on EMNLP: System Demonstrations*.

Olteanu, A.; Castillo, C.; Diaz, F.; and Vieweg, S. 2014. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *Proc. of AAAI*.

Petrović, S.; Osborne, M.; and Lavrenko, V. 2012. Using paraphrases for improving first story detection in news and Twitter. In *Proc. of NAACL*.

Potash, P.; Romanov, A.; and Rumshisky, A. 2017. SemEval-2017 Task 6: #HashtagWars: Learning a Sense of Humor. In *Proc. of SemEval*.

Ptáček, T.; Habernal, I.; and Hong, J. 2014. Sarcasm Detection on Czech and English Twitter. In *Proc. of COLING*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* .

Rajadesingan, A.; Zafarani, R.; and Liu, H. 2015. Sarcasm Detection on Twitter: A Behavioral Modeling Approach. In *Proc. of WSDM*.

Riloff, E.; Qadir, A.; Surve, P.; De Silva, L.; Gilbert, N.; and Huang, R. 2013. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In *Proceedings of the 2013 Conference on EMNLP*.

Rosenthal, S.; Farra, N.; and Nakov, P. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *Proc. of SemEval*.

Sarlin, P.-E.; DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2020. Superglue: Learning feature matching with graph neural networks. In *Proc. of CVPR*.

Thelwall, M.; Buckley, K.; and Paltoglou, G. 2012. Sentiment strength detection for the social web. *J. Assoc. Inf. Sci. Technol.* .

Thomas, J. A. 2014. *Meaning in interaction: An introduction to pragmatics*. Routledge.

Van Hee, C.; Lefever, E.; and Hoste, V. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. In *Proc. of SemEval*.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. of ICLR*.

Waseem, Z.; and Hovy, D. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proc. of NAACL*.

Xu, W.; Callison-Burch, C.; and Dolan, B. 2015. SemEval-2015 Task 1: Paraphrase and Semantic Similarity in Twitter (PIT). In *Proc. of SemEval*.

Zampieri, M.; Malmasi, S.; Nakov, P.; Rosenthal, S.; Farra, N.; and Kumar, R. 2019. Predicting the Type and Target of Offensive Posts in Social Media. In *Proc. of NAACL*.

Zubiaga, A. 2018. A longitudinal assessment of the persistence of twitter datasets. *J. Assoc. Inf. Sci. Technol.* 69(8).