7-2023

Impact of difficult negatives on Twitter crisis detection

Yuhao ZHANG
Singapore Management University, yuhaozhang@smu.edu.sg

Siaw Ling LO
Singapore Management University, sllo@smu.edu.sg

Phyo Yi WIN MYINT
Singapore Management University, ywmphyo@smu.edu.sg

Impact of Difficult Negatives on Twitter Crisis Detection

Completed Research Paper

Yuhao Zhang
Singapore Management University
80 Stamford Rd, Singapore 178902
yuhaozhang@smu.edu.sg

Siaw Ling Lo
Singapore Management University
80 Stamford Rd, Singapore 178902
sllo@smu.edu.sg

Phyo Yi Win Myint
Singapore Management University
80 Stamford Rd, Singapore 178902
ywmphyo@smu.edu.sg

Abstract

Twitter has become an alternative information source during a crisis. However, the short, noisy nature of tweets hinders information extraction. While models trained with standard Twitter crisis datasets accomplished decent performance, it remained a challenge to generalize to unseen crisis events. Thus, we proposed adding "difficult" negative examples during training to improve model generalization for Twitter crisis detection. Although adding random noise is a common practice, the impact of difficult negatives, i.e., negative data semantically similar to true examples, was never examined in NLP. Most of existing research focuses on the classification task, without considering the primary information need of crisis responders. In our study, we implemented multiple sequence tagging models and studied quantitatively and qualitatively the impact of difficult negatives on sequence tagging. We evaluated models on unseen events and showed that difficult negative forced models to generalize better, leading to more accurate information extraction in a real-world application.

Keywords: Twitter, Crisis Detection, Difficult Negative Data, Negative Mining

Introduction

Due to its open and real-time broadcasting nature, social media has become the go-to platform during incidents or crises. It serves as a valuable source of information, ranging from eyewitness accounts to seeking assistance. This study specifically concentrates on identifying social media content that can offer vital information on urban events, including crisis types (e.g., civil disorder, armed assault), location, the number of injuries or casualties, infrastructure damages, and weapon usage. Such information is crucial for first responders, such as the police force or paramedics, enabling them to effectively manage on-the-ground situations.

As a result, social media platforms like Twitter have become an alternative information source in crisis informatics. While some Twitter posts (tweets) provide valuable, first-hand information in crisis detection and monitoring, the majority of them are irrelevant, i.e., contain no crisis-related information. In a real-world crisis detection system, it is essential to detect new events unseen during the model training. Deep

Learning (DL)-based models can achieve state-of-the-art performance on Twitter-related tasks, e.g., text classification and Named Entity Recognition (NER), but they tend to memorize training examples and do not generalize well to unseen examples (Brigato and Iocchi 2020). The characteristics of tweets such as typos, abbreviations, and slangs make it more difficult to learn discriminative features.

In machine learning, negative examples (or negatives) are traditionally added to the positive examples (or positives) during training to make the model more robust and reduce the generalization error. Training examples can be generally categorised into "easy" and "difficult". The model can easily make correct predictions on easy examples, while it is more difficult on difficult ones. Both tweets in Figure 1 contain no crisis information (thus negatives) but the second one is more difficult to predict correctly due to words like "dying" and "fire" which appear frequently in crisis context. Despite causing more false predictions, difficult negatives have been proved to be effective for neural networks to learn discriminative features (Alon et al. 2019, Xuan et al. 2020).

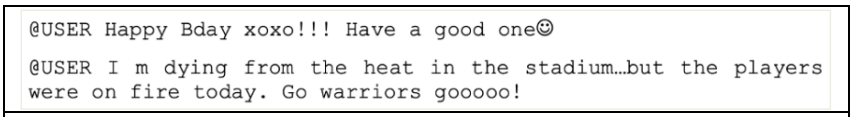

Figure 1. An Easy Negative Tweet (above) and a Difficult Negative Tweet (below)

Therefore, it is of interest to examine the impact of difficult negatives on the generalization capability of deep neural networks in Twitter crisis detection. While positives (crisis-related tweets) are typically obtained from a labelled dataset, the negatives have a very large selection space. The selection process of useful negatives is called negative mining. We proposed a semantic negative mining algorithm to select difficult negatives to be used during the training of several deep neural models. In order to examine the impact of difficult negatives on generalization, we evaluated models on separate hold-out datasets comprising of unseen examples. These examples, pertain to events not encountered during the training process, but fall within the six pre-defined crisis types examined in this study. Within this context, we sought to answer the following research questions:

1. What, if any, are the differences in prediction scores on test datasets comprising of unseen examples, depending on the addition of difficult negatives during training?
2. Can the result of Research Question 1 be extended to other sequence tagging models? In other words, can difficult negatives lead to better model generalization across different models?

For Research Question 1, we evaluated the impact of difficult negatives in various data settings, using no negatives and random negatives as baselines. We implemented a Bidirectional LSTM-CRF model as it is one of the most popular and effective sequence tagging models (Huang et al. 2015). Bidirectional Long Short-Term Memory (LSTM) can incorporate contexts from both forward and backward directions to represent the global information of the sequence. Conditional Random Field (CRF) is widely used as the inferencer for neural sequence tagging models as it considers the correlation between labels of adjacent words. We conducted an evaluation of the Precision, Recall, and F1 score on the test datasets, which consist of hold-out datasets and various types of negative datasets. Subsequently, we compared these results with the baselines to derive our conclusions. For Research Question 2, we implemented three popular sequence tagging models including LSTM-CRF, CNN-CRF and LSTM-CNN-CRF with different embeddings (GloVe, BERT, and BERTweet). Each model was trained in two training data settings: random negatives and difficult negatives. The model was subsequently evaluated on hold-out datasets containing the unseen examples before the conclusion was drawn. The main contributions of this paper are:

1. We proposed a semantic negative mining algorithm to select difficult negatives which were used during training to improve model generalization to unseen events in Twitter crisis detection.
2. We studied quantitatively the impact of difficult negatives on popular DL-based sequence tagging models by evaluating them on separate hold-out datasets.
3. We showed that negative examples had a significant impact on model generalization in Twitter crisis detection. Furthermore, we showed that difficult negatives improved the generalization of various DL-based sequence tagging models.

Terminology:

- Difficult negative: a negative example which is difficult for correct prediction due to its similarity to positive examples; often referred to as "hard negative" in Computer Vision.
- Easy negative: a negative example which is easy for correct prediction; random noise is often used as easy negatives to regulate overfitting.
- Hold-out datasets: separate test datasets consisting of unseen crisis events which belong to one of the pre-defined crisis classes. Predicting undefined classes at test time is the goal of Zero-Shot Learning and beyond the scope of this study.
- Anchor-negative similarity: similarity between a reference example (or anchor) and a negative example in the triplet loss function. The triplet loss function minimizes distances between points in the same class while maximizes distances between points from different classes. In this study, the anchor is selected as the average of positive examples. Therefore, anchor-negative similarity measures the similarity between a negative and an average positive.

Literature Review

Use of Negatives in Training and Negative Mining

Early studies in Natural Language Processing (NLP) have showed that adding random noise as negatives to training data improved model generalization as it prevented neural models from fitting individual data points precisely (Bishop 1995, Goodfellow et al. 1996). Random noise was also used in denoising autoencoders where the learning of robust feature extraction was aided by partially corrupting the training data on purpose (Vincent et al. 2008). Holmström and Koistinen (1992) used cross-validation to determine the optimal amount of noise to use during training. Greff et al. (2017) examined the effect of input noise, among other hyperparameters, on Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) model and found that additive random noise not only hurt performance but also increased training time. However, the model was not evaluated on separate hold-out datasets for generalization. In contrast to random noise which is often added as easy negatives, the use of difficult negatives and negative mining methods have not been substantially studied in NLP.

Negative mining has been used frequently in Computer Vision and deep metric learning where a contrastive loss or triplet loss is optimized. The goal is to make data points in the same class closer to each other than those from a different class. The triplet loss achieves this by comparing a reference point (called "anchor") to a positive and a negative point. The training objective is to maximize anchor-positive similarity (*Sap*) and minimize anchor-negative similarity (*San*). Xuan et al. (2020) defined difficult training triplets (anchor, positive, negative) as triplets whose *San* is equal to or higher than *Sap*. They showed that difficult examples led to more generalizable features in image retrieval task. Due to the great success of deep metric learning in Computer Vision, various negative mining methods have been proposed to select optimal difficult negatives to be used in training (Schroff et al. 2015, Huang et al. 2016, Manmatha et al. 2017, Harwood et al. 2017, Robinson et al. 2020, Vasudeva et al. 2021).

In general, there are two types of negative mining methods: model-based and statistical approach. The model-based approach first trains a model to predict all negative samples and selects false negatives as high-quality negative samples. The model is subsequently re-trained and the process may require a few more iterations. The intuition is that the model re-trained with these negative samples should perform better with the extra knowledge and predict less false positives. The disadvantage with model-based negative mining is the time consumption as it is necessary to predict and find the most valuable negative samples on all negative samples for a few iterations of training. One alternative is online negative mining, in which negative samples with a larger loss for gradient update were selected to update the model (Shrivastava et al. 2016). This, however, has only been studied in the field of Compute Vision.

For statistical approach, negative samples are selected based on some statistical measure. For example, in our study, we used the cosine similarity measure between the anchor and negative examples (anchor-negative similarity or *San*). The anchor is the average representation of positive examples. Therefore, the higher the *San*, the more difficult a negative example is. This type of negative examples can force the model learn the semantic information in the text, rather than learning the literal meaning. The downside of statistical approach is that, compared to the model-based approach, selected negative samples may not be the samples that can best guide the model to the direction of the steepest gradient descent.

While most studies on negative mining were in the field of Computer Vision, the idea of difficult negatives can be equally applied in NLP. Alon et al. (2019) trained a contextualized neural speech recognition model by adding phonetically similar phrases as difficult negatives. Although the impact was not studied quantitatively, they conducted a qualitative analysis to show that difficult negatives led to better discrimination of subtle phonetic differences. To the best of our knowledge, no prior NLP research has used semantically similar tweets as difficult negatives and studied their impact on Twitter crisis detection.

Twitter Crisis Detection

Social media crisis informatics has greatly benefited from the advancement of deep learning. Nguyen et al. (2017) implemented a Convolutional Neural Network (CNN) model for crisis tweet classification and showed that it outperformed non-neural models in both in-domain and out-of-domain datasets. Madichetty and Sridevi (2020) classified crisis information types using a dense neural model which outperformed baseline Support Vector Machine and CNN models. However, the study was limited to in-domain data. Paul et al. (2021) proposed a state-of-the-art hybrid neural network model for Twitter crisis detection. The study again used only an in-domain dataset comprising a small number of events. Therefore, it remains unknown how well these models trained on one dataset generalize to unseen events. Furthermore, most Twitter crisis detection models formulated the problem as a text classification task while few proposed a sequence tagging approach. Sequence tagging can provide a second level of actionable details beyond classification to better address the primary information needs of crisis responders (Zade et al. 2018). For example, Figure 2 shows a positive crisis tweet with relevant information tagged as bolded, underlined with labels (in squared bracket). These details offer critical insights for crisis responders that go beyond a simple positive classification, enabling them to gain essential knowledge and make informed decisions. A typical neural sequence tagging model consists of three parts: embedding module, context encoder and inferencer (He et al. 2020). The embedding module utilizes a pretrained word embedding to map words into their distributed representations as the initial input of the model. The context encoder extracts contextual features and dependencies of an input sequence and passes the learned features into inferencer for label prediction. LSTM and CNN are the most popular context encoders. Bidirectional LSTM can incorporate contexts from both forward and backward directions to generate the hidden states of each token, and then represent the global information of the sequence. CNN can extract local and hierarchical features and is more computationally efficient than LSTM, but it has difficulties in capturing long-range dependencies. Conditional Random Field (CRF) (Lafferty et al. 2001) is widely used as the inferencer for most neural sequence tagging models as it considers the correlation between labels of adjacent words (Huang et al. 2015, Ma and Hovy 2016, Rei 2017). In this study, the effectiveness of combining LSTM, CNN, and CRF as a sequence tagger was tested. Detailed information on the model architecture can be found in the Method - Model Architecture subsection.

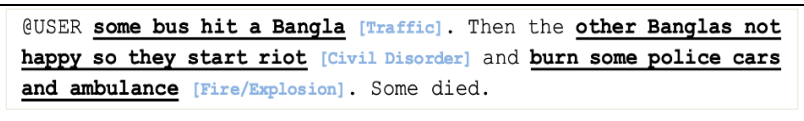

Figure 2. A Positive Example with Labels

Method

Problem Formulation

We modeled Twitter crisis detection as a sequence tagging task. A sequence tagging task takes a sequence of tokens $x = (x_1, \ldots x_n)$ as the input and predicts a label for each token. The output is sequence $y = (y_1, \ldots y_n)$, where each y_i is the label of x_i. Sequence tagging usually leads to lower performance than classification

on the same dataset as it is more difficult to predict each token than the entire text[1]. Six crisis types were identified based on the practical need of crisis response in urban context (Table 1).

A span is labelled with corresponding crisis types if it contains actionable details of a crisis such as "what", "who", "when" and "where". For example, the span "some bus hit a Bangla [Traffic]" (Figure 2) mentioned a traffic incident involving a bus and a Bangladesh national. In order to produce more cohesive spans, we included intermediate tokens if the resulting span was grammatically complete. For instance, we labelled "other Banglas not happy so they start riot" instead of "other Banglas" and "start riot" separately. In addition, although all crisis tweets have a main crisis type, they can contain spans of other crisis types. The crisis tweet in Figure 2 was from 2013 Singapore Little India Riot, therefore a "Civil Disorder" crisis, but it also contained "Traffic" and "Fire/Explosion" spans. This way, the complexity of the event could be better understood, compared to labeling all spans in a given tweet with one crisis type.

Crisis Tweets

We collected 864 crisis tweets (positive examples) from data sources including CrisisLexT26 (Olteanu et al. 2015), Traffic Tweets (Dabiri and Heaslip 2019), Disasters on Social Media (Crowdflower 2015), TREC-IS (McCreadie et al. 2020), Civil Unrests on Twitter (Sech et al. 2020) and Twitter API. In order to include more linguistic variations in the training data for better generalization, multiple events were collected for each crisis type before tweets were annotated with crisis spans.

Crisis Type (size)	Crisis Event (size)
Traffic (146)	mixed traffic crashes (50), 2013 Glasgow Helicopter Crash (48), 2013 NYC Train Crash (48)
Fire/Explosion (145)	2013 West Texas Explosion (43), 2013 Brazil Nightclub Fire (46), 2012 Colorado Wildfires (36), 2019 Durham Gas Explosion (20)
Flood/Typhoon (141)	2012 Typhoon Pablo (49), 2013 Alberta Floods (34), 2013 Typhoon Yolanda (38), 2020 Edenville Dam Failure (20)
Civil Disorder (146)	mixed civil unrests from 42 countries (87), 2013 Singapore Little India Riot (39), 2020 U.S. Capitol Riot (20)
Shooting (143)	2013 LA Airport Shootings (34), 2020 South Carolina Bar Shooting (35), 2020 Texas University Shooting (35), 2018 Pittsburgh Synagogue Shooting (39)
Bombing (143)	2016 Brussels Bombings (34), 2017 Manchester Arena Bombing (31), 2013 Boston Bombings (45), mixed bombings on social media (33)

Table 1. Twitter Crisis Types and Events

Negative Mining

To study the impact of difficult negatives, we proposed an unsupervised, semantic negative mining algorithm. The underlying assumption was that negatives with high semantic similarity to the anchor (measured by anchor-negative similarity) were more likely to be predicted as false positives, therefore can be used as difficult negatives in the training. This enabled us to search them in a large unlabelled corpus. The algorithm comprised of three steps (Figure 3): (1) calculate the anchor vector from positive examples;

[1] It was worth mentioning that the leading F1 score of Twitter NER (a sequence tagging task) is 59.5% on WNUT 2016 NER dataset: https://paperswithcode.com/sota/named-entity-recognition-on-wnut-2016.

(2) randomly sample *p* negatives to compute anchor-negative similarities (or *San*) and fit a normal distribution to the frequency distribution of *San*; (3) iterate negatives one by one, calculate its *San* and select difficult negatives by the position of their *San* in the normal distribution.

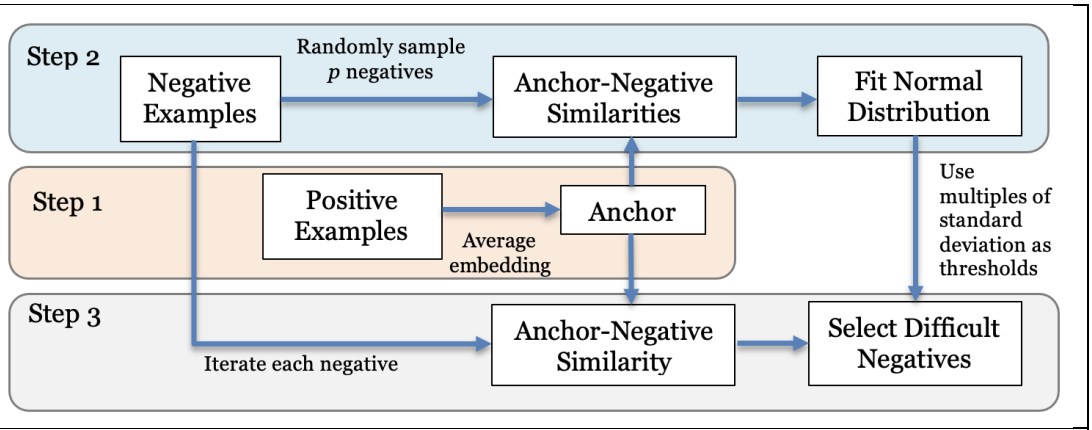

Figure 3. Workflow Diagram of Semantic Negative Mining

Calculate Anchor Vector

We utilized a pretrained Sentence Transformer (Reimers and Gurevych 2019) to represent tweets in the embedding space and used the cosine similarity to measure the semantic similarity. Specifically, we used "all-mpnet-base-v2"[2] as it was trained for semantic search and suitable for the given task. Anchors were calculated as average embedding vectors of positives in each crisis type. Principle Component Analysis (PCA) plot showed the distribution of positives and anchors in reduced dimensions[3] (Figure 4). It can be observed that data points in the same class were close to each other than points from a different class. The anchors were centre points in each crisis type cluster (shown as red circles).

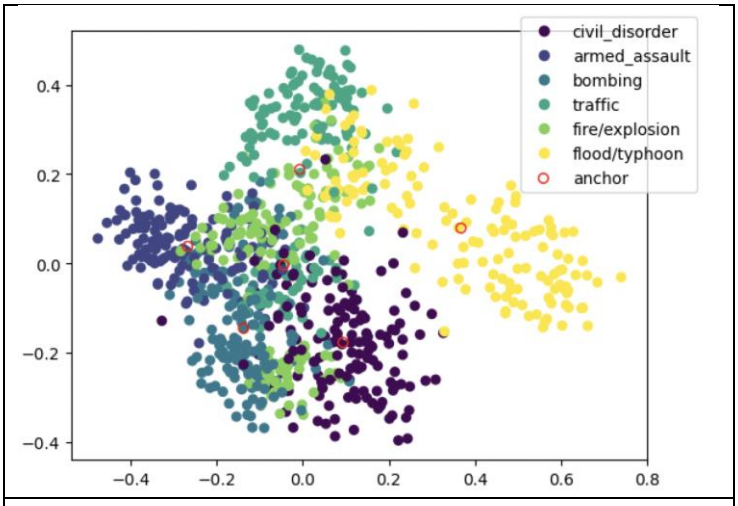

Figure 4. PCA Plot of Positives and Anchors

[2] https://www.sbert.net/docs/pretrained_models.html

[3] The fist two principal components were shown in Figure 4. They only accounted for 6.88% and 4.91% of the explained variation respectively.

Fit Normal Distribution

Next, we sampled half a million of random tweets using Twitter API. This large unlabeled corpus (denoted as *N)* contained mostly irrelevant tweets therefore can be viewed approximately as negatives. We then sampled p (e.g., 3000) negatives from N to plot the frequency distributions of *San* for each crisis type. As shown in Figure 5, all *San* distributions can be approximated by normal distributions with slightly different means μ and standard variations σ (or sigma).

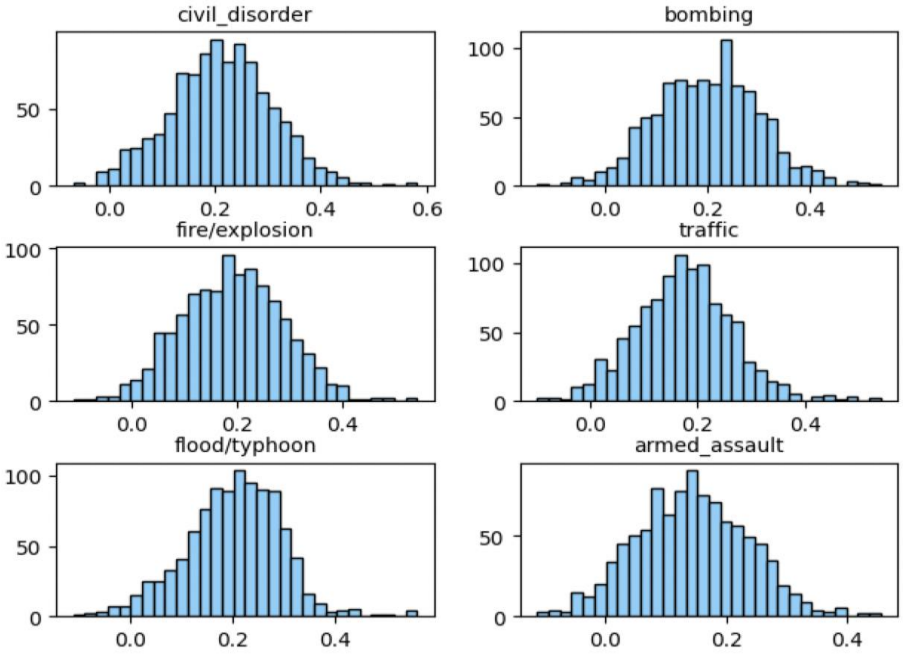

Figure 5. Frequency Distribution of Anchor-Negative Similarity Follows Normal Distribution

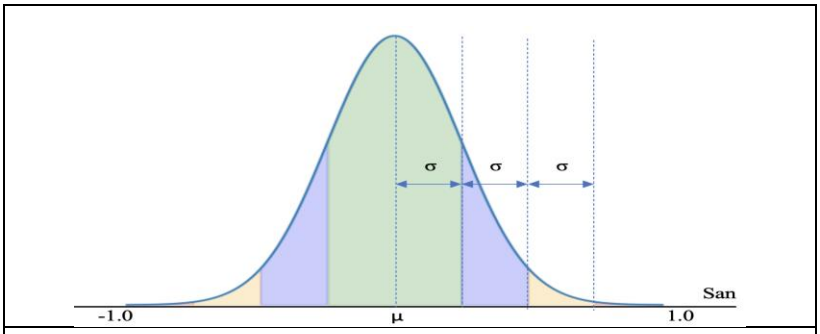

Figure 6. We Defined Difficulty Levels Using Multiples of Standard Deviation in a Normal Distribution.

Search Negatives by the Position of San in Normal Distribution

A high *San* entails a high semantic similarity between a negative and an anchor (average of positives), therefore more difficult when it comes to model prediction. Consequently, the negatives on two ends of a normal distribution are more difficult than those near the center (Figure 6). We used multiples of sigmas to divide the area under the curve into three regions. The green area corresponds to easiest negatives (denoted as one-sigma or 1σ). The blue corresponds to semi-difficult negatives (2σ). The yellow area to the most difficult negatives (3σ). We did not consider negatives beyond three sigmas as they only account for

0.3% of the data. We searched negatives with three difficulty levels (1σ, 2σ, 3σ) by running the algorithm (Figure 7) for each crisis type[4] and combined them with positives.

```
Algorithm SemanticNegativeMining(C, N, k, p)
  Input: crisis tweets C, negative tweets N, output size k, sampling
         size p to calculate normal distribution
  Output: one-sigma negative tweets E, two-sigma negative tweets S,
         three-sigma negative tweets D
E, S, D = { }, { }, { }
// Step 1: Calculate anchor vector.
// h(x) is the word embeddings of tweet x
anchor vector h(a) = Average[h(c)] for all crisis tweet c in C
// Step 2: Fit a normal distribution to San distribution
San = { }
for each in p negative tweets randomly sampled from N
    San.append(cosine-similarity(h(a), h(each)))
San ~ Normal Distribution (μ, σ)
// Step 3: Assign negative data to each group according to the
// position of San in the normal distribution
while any(length(E, S, D) < k):
    sample a negative tweet n from N
    similarity = cosine-similarity(h(a), h(n))
    if length(E) < k and μ-σ <= similarity <= μ+σ
        E.append(n)
    else if length(S) < k and ((μ-2*σ <= similarity <= μ-σ)
                    or (μ+σ <= similarity <= μ+2*σ))
        S.append(n)
    else if length(D) < k and ((μ-3*σ <= similarity <= μ-2*σ)
                    or (μ+2*σ <= similarity <= μ+3*σ))
        D.append(n)
return E, S, D
```

Figure 7. Semantic Negative Mining Algorithm

Datasets

Positives and negatives were combined to form train and test datasets (Table 2). For 864 crisis tweets in the train, we collected 864 negative tweets for each difficult level. They are denoted as $\text{train}_{[1\sigma]}$, $\text{train}_{[2\sigma]}$, and $\text{train}_{[3\sigma]}$. In addition, we included no negatives ($\text{train}_{[no]}$) and random negatives ($\text{train}_{[random]}$) as baseline train datasets.

Dataset	Description
$\text{train}_{[no]}$	864 crisis tweets
$\text{train}_{[random]}$	864 crisis tweets + 864 randomly sampled negative tweets
$\text{train}_{[1\sigma]}$, $\text{train}_{[2\sigma]}$, $\text{train}_{[3\sigma]}$	864 crisis tweets + 864 negative tweets of various difficulty levels
$\text{test}_{[random]}$	269 unseen crisis tweets + 269 randomly sampled negative tweets
$\text{test}_{[random+]}$	269 unseen crisis tweets + 2690 randomly sampled negative tweets
$\text{test}_{[1\sigma]}$, $\text{test}_{[2\sigma]}$, $\text{test}_{[3\sigma]}$	269 unseen crisis tweets + 269 negative tweets of various difficulty levels

Table 2. Train and Test Datasets

[4] Semantic search for 864 difficult negatives took 48, 113, and 632 seconds respectively for 1σ, 2σ, and 3σ using CPU on Intel Xeon Gold 6342 Processor @2.80GHz

For evaluation, we collected and annotated 269 additional crisis tweets (Table 3). These unseen crisis tweets were then combined with negatives to form five different test settings. The difficult negatives were selected using our algorithm to form test $_{[1\sigma]}$, test $_{[2\sigma]}$, and test $_{[3\sigma]}$. In addition, we created test$_{[random]}$ with 269 randomly sampled negatives and test$_{[random+]}$ with 2690 randomly sampled negatives. The test$_{[random+]}$ had a positive-negative ratio of 1:10 as it was used to evaluate how well the model generalized in the real-world scenario where the majority of tweets were irrelevant. The remaining test datasets had a positive-negative ratio of 1:1.

Crisis Type (size)	Crisis Event (size)
Traffic (45)	2013 Lac-Megantic Train Crash (45)
Fire/Explosion (44)	2016 Puttingal Temple Explosion (22), 2017 Lilac Wildfire (22)
Flood/Typhoon (45)	2013 Queensland Floods (23), 2018 Hurricane Florence (22)
Civil Disorder (45)	2022 Iran Protests (45)
Shooting (44)	2017 Dallas Shooting (44)
Bombing (46)	2015 Paris Attacks (46)

Table 3. Crisis Events in Hold-out Datasets

Model Architecture

We implemented several popular sequence tagging models including LSTM-CRF, CNN-CRF, and LSTM-CNN-CRF due to their effectiveness and popularity in the field. All LSTM models had one forward and one backward layer, with a hidden dimension of 128. The LSTM dropout was set to 0.4. The kernel size of CNN was set to 3. For feature extraction, we used BERTweet (Nguyen et al. 2020), a transformer-based Language Model trained on English tweets which achieved state-of-the-art performance in Twitter-related tasks. For comparison, we also included BERT (Devlin et al. 2019) and GloVe (Pennington et al. 2014) embeddings. The specific versions for embeddings were "vinai/bertweet-base", "bert-base-cased", and "en-twitter" respectively. The maximum sequence length was set to 64. The remaining parameters followed default configurations.

Experiment

Two experiments were conducted to address the two Research Questions (RQs) defined at the beginning. Exp. 1 aimed to find out which negative data setting led to the best generalization on BERTweet-LSTM-CRF model. If the difficult negatives improved model generalization compared to no negatives or random negatives, we then examined whether this effect held true for other sequence tagging models in Exp. 2.

Tweet Pre-processing

BERTweet employed a normalization strategy in which twitter-specific tokens of user mentions and web links were transformed into special tokens "@USER" and "HTTPURL". No additional tweet pre-processing was done except for converting emoticons to corresponding texts. The same pre-processing was applied for BERT and GloVe embeddings.

Training Configurations

All models were trained with AdamW optimizer and a linear scheduler without warmups. For Exp.1, we found through grid search[5] that BERTweet-LSTM-CRF achieved the best F1 with the learning rate of 2e-5 and the mini batch size of 4. For Exp.2, we did not finetune the hyperparameters for each model individually due to time constraint. Instead, we used mini batch size of 16 for all models. We used a learning rate of 0.1 for the GloVe model and 2e-5 for BERT and BERTweet models. For all experiments, training would stop if

[5] We conducted a grid search over learning rates [2e-4, 1e-4, 5e-5, 2e-5, 1e-5] and mini batch sizes of [1, 2, 4, 8, 16, 32] using 5-fold cross-validation on train$_{[random]}$ dataset.

the micro F1 score on the test split did not improve for 4 epochs or if the maximum number of 40 epochs was reached. All results were averaged using 5-fold cross-validation.

Exp. 1 to address RQ 1: Do Difficult Negatives Improve Generalization?

Since Exp. 1 studied the impact of negatives, the only changing variable was the type of negatives in the train datasets, i.e., no negatives, random negatives and three levels of difficult negatives: 1σ, 2σ and 3σ. Table 4 shows the F1 scores[6] of BERTweet-LSTM-CRF in different data settings. Train dataset with 2σ difficult negatives had the best F1 scores over all test settings. When evaluated on $test_{[random+]}$ which had the most realistic positive-negative ratio of 1:10, $train_{[2\sigma]}$ outperformed $train_{[no]}$ by 11.28% and $train_{[random]}$ by 1.54%. Our result showed that adding negatives had a significant impact on generalization in sequence tagging task. The same effect has been shown by previous researchers in the text classification task (Bishop 1995, Goodfellow et al. 1996). In addition, difficult negatives can further improve the generalization over random negatives. Both 1σ and 2σ negatives outperformed random negatives in all test settings. This showed that difficult negatives can indeed improve generalization. Compared to $train_{[random]}$, $train_{[2\sigma]}$ improved Precision by 1.07%, Recall by 2.26%, and F1 by 1.54% on on $test_{[random+]}$ (Table 5). Lastly, from the number of epochs trained, we can see that difficult negatives helped the model converge faster (last column of Table 4).

$train_{[x]}$	$test_{[x]}$					epoch
	random	random+	1σ	2σ	3σ	
no	40.40	27.65	40.27	39.97	37.19	22.2
random	40.29	37.39	40.23	39.46	38.80	20.2
1σ	41.77	38.37	41.71	41.25	40.04	18.4
2σ	**41.96**	**38.93**	**41.96**	**41.50**	**40.36**	19
3σ	40.23	38.61	40.21	39.88	39.47	19.6

Table 4. LSTM-CRF F1 Scores in Different Data Settings

$test_{[x]}$	$train_{[x]}$	Precision	Recall	F1
random+	random	32.82	43.50	37.39
	2σ	**33.89**	**45.76**	**38.93**

Table 5. Precision, Recall and F1 Scores

Exp. 2 to address RQ 2: Is the Impact of Difficult Negatives Consistent Across Multiple Models?

Since $train_{[2\sigma]}$ had the best F1 scores over all test datasets in Exp. 1, we used it to train other sequence tagging models to find out whether the impact would be consistent.

The effect of difficult negatives can be observed across several sequence tagging models[7] (Table 6) though at various degrees. The improvement in F1 scores for BERTweet models ranged from 1.23% to 3.79%. The improvement for BERT models ranged from 0.38% to 0.68%. The improvement for GloVe models ranged from 0.09% to 0.45%. The difference in improvement might be due to several reasons: (1) the best difficult negatives (2σ) found in Exp. 1 based on BERTweet-LSTM-CRF might not be optimal for other models here; (2) hyperparameters were not finetuned for each individual model. These factors such as batch size can

[6] F1 scores on the test split were in the range of 63-67% for all train datasets. The performance dropped sharply on separate test datasets because they comprised of unseen crisis events with different feature space.

[7] We only implemented several popular sequence tagging architectures with proven success in the field. Therefore, not all combinations were implemented.

affect the degree of improvement by difficult negatives. In this study, we assessed the impact of difficult negatives across multiple model architectures and observed their improved generalization property. Nevertheless, as demonstrated in Table 5 and Table 6, conducting further experiments and optimizing parameter settings are necessary to identify the most effective model architecture and data setting for a specific problem.

Word Embedding	Model	$test_{[random+]}$		
		$train_{[2\sigma]}$	$train_{[random]}$	diff
GloVe	LSTM	**12.96**	12.87	+0.09
GloVe	LSTM-CRF	**25.25**	24.80	+0.45
BERT	CNN-CRF	**34.57**	33.89	+0.68
BERT	LSTM-CRF	**34.52**	33.88	+0.64
BERT	LSTM-CNN-CRF	**31.95**	31.57	+0.38
BERTweet	CNN-CRF	**37.64**	36.41	+1.23
BERTweet	LSTM-CRF	**37.19**	33.40	+3.79
BERTweet	LSTM-CNN-CRF	**35.48**	34.02	+1.46

Table 6. F1 Scores on $test_{[random+]}$ by Various Sequence Taggers Trained by Difficult and Random Negatives

Qualitative Analysis

We present a qualitative analysis to show that difficult negatives helped the model to better understand the semantic difference. In Figure 8, the model trained with random negatives made false positive predictions (marked in red) on negative tweets. The first mistake was likely because the word "flocking" (meaning "move together as a crowd") was related to the civil disorder context, therefore confused the model. The false predictions in the second example were likely due to overfitting to the association of "plant" and "explosion" in the train data (i.e., West Texas fertilizer plant explosion). The model trained with difficult negatives, on the other hand, correctly recognized both examples as negatives, making no crisis predictions.

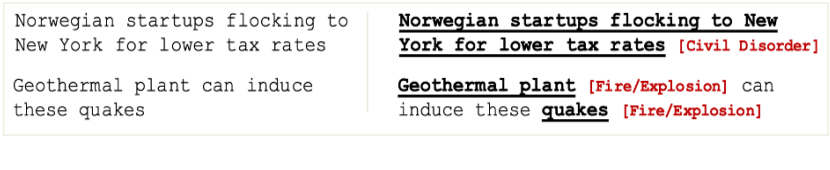

Figure 8. Predictions on Negative Tweets by Models Trained with Difficult (left) and Random (right) Negatives

Similar observations were made in positive tweets (see Figure 9). While both models correctly predicted one span (marked in green), the model trained with random negatives made an additional false prediction due to overfitting to the word "crashes" in "Traffic" tweets. In comparison, difficult negatives forced the model to learn more robust features, therefore alleviating the overfitting issue. We believe that this finding is helpful for field practitioners who wish to deploy a Twitter crisis detection model in a real-world application.

WATCH: House washed away by **flooding river in southern Canada** [Flood/Typhoon] crashes into bridge	WATCH: House washed away by **flooding river in southern Canada** [Flood/Typhoon] **crashes into bridge** [Traffic]

Figure 9. Predictions on a Crisis Tweet by Models Trained with Difficult (left) and Random (right) Negatives

Future Work

For future work, we plan to perform additional experiments on other sequence tagging tasks such as NER and Semantic Role Labeling. Examining the impact of difficult negatives on Twitter classification tasks, using publicly available Twitter crisis datasets such CrisisLexT26 and TREC-IS, is also in the discussion. We plan to improve the negative mining algorithm to find the optimal difficult negatives for NLP tasks in order to best guide the model training. Furthermore, it would be interesting to analyze how training configurations such as mini batch size and learning rate can affect the impact of difficult negatives. For example, in Exp 1, we used grid search to determine that the mini batch size of 4 yielded the optimal performance for LSTM-CRF model. However, when we evaluated the model at different batch sizes, the level of improvement by difficult negatives varied (Figure 10). At batch size 4, train$_{[2\sigma]}$ improved F1 by 1.54%. At batch size 16, the improvement was increased 3.79%. At batch size 2 and 32, difficult negatives led to worse performance. This suggested that the level of improvement by difficult negatives was influenced by the batch size, among other factors. Therefore, further experiments are necessary to determine the optimal parameter settings for a specific model and problem.

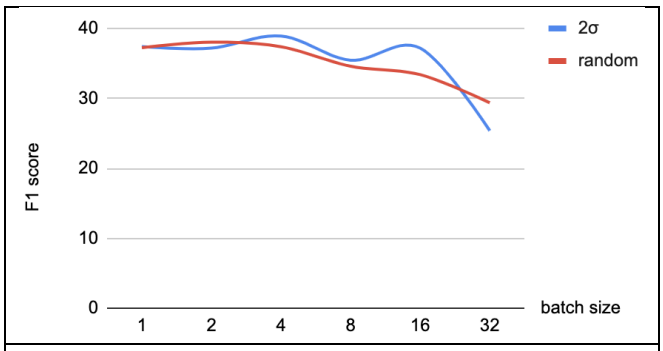

Figure 10. F1 Scores on test$_{[random+]}$ by train$_{[2\sigma]}$ and train$_{[random]}$ at Different Batch Sizes

The aim of this paper is to assess the impact of difficult negatives in Twitter crisis detection so that a practical model can be developed and deployed in the real-world to aid the decision-making process of crisis responders. From our results, it is obvious that difficult negatives help in model generalization but at the same time, it also highlighted the challenge of extracting relevant and insightful content from social media. It is worth noting that sequence tagging tasks often exhibit lower performance on the same dataset compared to classification tasks. This is primarily due to the increased difficulty of predicting a label for each individual token in the sequence. For example, the leading F1 score on the test split of WNUT 2016 NER dataset is only 59.5%[8]. As a comparison, we did additional experiments and the F1 scores on test split (20% of train dataset) for all BERTweet-LSTM-CRF models trained with different data settings were in the range of 63-67% via 5-fold cross validation. In this study, we modeled the crisis detection as a sequence labeling task, where the crisis sequence was a span of word tokens that contain crisis details such as entities

[8] https://paperswithcode.com/sota/named-entity-recognition-on-wnut-2016.

involved ("what" and "who"), locations ("where") and action words ("how") (see Figure 2). Compared to simply classifying tweets (e.g., "Traffic" or "Bombing") or capturing keywords (e.g., "crashed", "smashing", "burning"), sequence labeling can provide a second level of information, though at the cost of lower prediction scores. One possible enhancement is to introduce a new labeling scheme that differentiate the ACTOR (people or organization involved); ACTION; CRISIS (crisis names) etc. so that the details can be labelled by their roles (Figure 11) instead of identifying a span (Figure 2). This may potentially achieve a better performance due to its finer grain with more distinctive part of speech recognition.

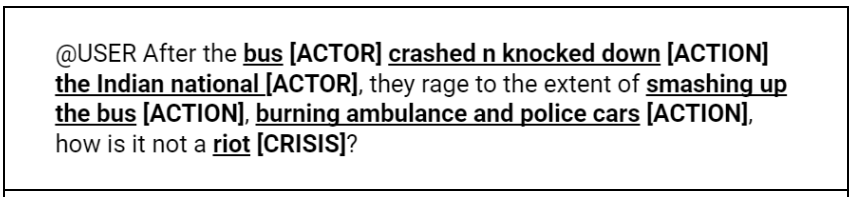

Figure 11. A Suggested New Labeling Scheme

Conclusion

In this paper, we proposed an unsupervised, semantic negative mining algorithm to select difficult negative data and used it in training to improve model generalization in Twitter crisis detection. We studied quantitatively the impact of difficult negatives by implementing several popular sequence tagging models and evaluated them under various data settings. By evaluating on separate hold-out datasets, we showed that difficult negatives led to better generalization and the effect was consistent over all implemented models. Lastly, we conducted a qualitative analysis to demonstrate that the model trained with difficult negatives was able to learn more robust features and have less false positives.

Acknowledgements

The Authors would like to acknowledge the support and project funding from ST Engineering Mission Software & Services Pte Ltd under Research Collaboration Agreement No: 001052-00001.

References

Alon, U., Pundak, G., & Sainath, T. N. (2019). Contextual Speech Recognition with Difficult Negative Training Examples. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6440-6444).

Bishop, C. (1995). *Neural Networks for Pattern Recognition* (1st ed.). Oxford University Press, USA.

Brigato, L., & Iocchi, L. (2020). A Close Look at Deep Learning with Small Data. In *25th International Conference on Pattern Recognition (ICPR)* (pp. 2490-2497).

Crowdflower. (2015). *Disasters on Social Media*. data.world. https://data.world/crowdflower/disasters-on-social-media.

Dabiri, S., & Heaslip, K. (2019). Developing a Twitter-based traffic event detection model using deep learning architectures. *Expert Systems with Applications, 118*, 425-439.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/arXiv.1810.04805

Goodfellow, I., Bengio, Y., & Courville, A. (1996). *Deep Learning (Adaptive Computation and Machine Learning series)*. Oxford University Press, USA.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems, 28* (10), 2222-2232.

Harwood, B., Kumar, B.G.V., Carneiro, G., Reid, I., & Drummond, T. (2017). Smart mining for deep metric learning. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 2840–2848).

He, Z., Wang, Z., Wei, W., Feng, S., Mao, X., & Jiang, S. (2020). A survey on recent advances in sequence labeling from deep learning models. https://doi.org/10.48550/arXiv.2011.06727

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.

Holmström, L., & Koistinen, P. (1992). Using additive noise in back-propagation training. *IEEE Transactions on Neural Networks, 3*(1), 24-38.

Huang, C., Loy, C., & Tang, X. (2016). Local similarity-aware deep feature embedding. In *Advances in Neural Information Processing Systems (NeurIPS)* (pp. 1262-1270).

Huang, Z., Wei X., & Kai, Y. (2015). Bidirectional LSTM-CRF models for sequence tagging. https://doi.org/10.48550/arXiv.1508.01991

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *18th International Conference on Machine Learning*.

Ma, X., & Hovy, E. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (pp. 1064-1074)

Madichetty, S., & Sridevi M. (2020). Improved Classification of Crisis-Related Data on Twitter using Contextual Representations. *Procedia Computer Science, 167*, 962-968.

Manmatha, R., C. Wu, C., Smola, A., & Krahenbuhl, P. (2017). Sampling matters in deep embedding learning. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 2859-2876).

McCreadie, R., Buntain, C., & Soboroff, I. (2020). Incident Streams 2019: Actionable Insights and How to Find Them. In *17th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2020)* (pp. 744–760).

Nguyen, D. Q., Vu, T., & Nguyen, A. T. (2020). BERTweet: A pre-trained language model for English Tweets. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 9–14).

Nguyen, D.T., Al Mannai, K. A., Joty, S., Sajjad, H., Imran, M., & Mitra, P. (2017). Robust classification of crisis-related data on social networks using convolutional neural networks. *Proceedings of the 11th International AAAI Conference on Web and Social Media*.

Olteanu, A., Vieweg, S., & Castillo, C. (2015). What to Expect When the Unexpected Happens: Social Media Communications Across Crises. *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*.

Paul, N. R., Sahoo, M., Hati, S. K., & Sahoo, T. (2021). Detecting Disaster Related Tweets Using Hybrid Deep Neural Network Models. *Proceedings of 2021 International Conference on Advances in Technology, Management & Education (ICATME)* (pp. 71-76).

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532-1543).

Rei, M. (2017). Semi-supervised Multitask Learning for Sequence Labeling. https://doi.org/10.48550/arXiv.1704.07156

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERTNetworks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Robinson, J., Chuang, C., Sra, S., & Jegelka, S. (2020). Contrastive Learning with Hard Negative Samples. https://doi.org/10.48550/arXiv.2010.04592

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 815-823).

Sech, J., DeLucia, A., Buczak, A. L., & Dredze, M. (2020). Civil Unrest on Twitter (CUT): A Dataset of Tweets to Support Research on Civil Unrest. *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)* (pp. 215–221).

Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with online hard example mining, *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2016.89

Vasudeva, B., Deora, P., Bhattacharya, S., Pal, U., & Chanda, S. (2021). Loop: Looking for optimal hard negative embeddings for deep metric learning. *Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/iccv48922.2021.01046

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th international conference on Machine learning (ICML)* (pp. 1096–1103).

Xuan, H., Stylianou, A., Liu, X., & Pless, R. (2020). Hard negative examples are hard, but useful. *Proceedings of Computer Vision – ECCV 2020* (pp. 126–142). https://doi.org/10.48550/arXiv.2007.12749

Zade, H., Shah, K., Rangarajan, V., Kshirsagar, P., Imran, M., & Starbird, K. (2018). From Situational Awareness to Actionability: Towards Improving the Utility of Social Media Data for Crisis Response. *Proceedings of the ACM on Human-Computer Interaction, 2 (CSCW)* (pp. 1–18).