CrisisTransformers: Pre-trained language models and sentence encoders for crisis-related social media texts

Rabindra Lamsal[a,*], Maria Rodriguez Read[a], Shanika Karunasekera[a]

[a]*School of Computing and Information Systems, The University of Melbourne, Australia*

Abstract

Social media platforms play an essential role in crisis communication, but analyzing crisis-related social media texts is challenging due to their informal nature. Transformer-based pre-trained models like BERT and RoBERTa have shown success in various NLP tasks, but they are not tailored for crisis-related texts. Furthermore, general-purpose sentence encoders are used to generate sentence embeddings, regardless of the textual complexities in crisis-related texts. Advances in applications like text classification, semantic search, and clustering contribute to effective processing of crisis-related texts, which is essential for emergency responders to gain a comprehensive view of a crisis event, whether historical or real-time. To address these gaps in crisis informatics literature, this study introduces *CrisisTransformers*, an ensemble of pre-trained language models and sentence encoders trained on an extensive corpus of over 15 billion word tokens from tweets associated with more than 30 crisis events, including disease outbreaks, natural disasters, conflicts, and other critical incidents. We evaluate existing models and CrisisTransformers on 18 crisis-specific public datasets. Our pre-trained models outperform strong baselines across all datasets in classification tasks, and our best-performing sentence encoder improves the state-of-the-art by 17.43% in sentence encoding tasks. Additionally, we investigate the impact of model initialization on convergence and evaluate the significance of domain-specific models in generating semantically meaningful sentence embeddings. All models are publicly released (`https://huggingface.co/crisistransformers`), with the anticipation that they will serve as a robust baseline for tasks involving the analysis of crisis-related social media texts.

*rlamsal@student.unimelb.edu.au

Keywords: classification models, sentence encoding models, crisis informatics, social media analytics, social computing

1. Introduction

Social media platforms, such as Facebook and Twitter, have become an essential medium for information sharing and communication during times of crisis [1, 2]. Particularly during disasters, such as wildfires, earthquakes, hurricanes, tsunamis, floods, cyclones, and epidemics, social media platforms play a pivotal role in the timely dissemination of information. These platforms are critical information sources for affected individuals and emergency responders, enabling real-time updates on evolving situations and providing firsthand accounts from those directly and indirectly impacted. In general, social media contributes to community mobilization, i.e. enabling individuals to seek and offer assistance and organize relief efforts. The enormous amount of user-generated content on social media platforms acts as a rich source of historical as well as real-time data. However, the volume and textual complexity of crisis-related social media texts give rise to multiple challenges for effective analysis and understanding. The volume necessitates automated analysis as the number of conversations increases exponentially during a crisis, and the textual complexity involves dealing with informally written texts with a significant presence of acronyms, misspellings, hashtags, mentions, etc.

Domain-specific pre-trained language models have helped produce state-of-the-art results for numerous NLP tasks in various areas such as biomedical research [3], scientific literature analysis [4], clinical text analysis [5] and financial text analysis [6]. Trained on massive amounts of domain-specific texts, these models produce contextual text representations within their respective domains. Likewise, the potential of transformer-based [7] pre-trained models can be harnessed to understand and analyze crisis-related social media texts for effective and efficient crisis response and management. Despite the broad array of domains in which pre-trained models have been employed, a notable gap exists, i.e., the absence of pre-trained models explicitly tailored for crisis-related social media texts. Furthermore, pre-trained language mod-

els do not produce semantically rich sentence embeddings[1], critical for tasks like semantic search and clustering [8]. Currently, the generation of semantically meaningful sentence embeddings, regardless of the domain, relies on general-purpose sentence embedding models (sentence encoders) [8, 9]. These models utilize pre-trained models that have been trained on corpora comprising texts from broad and general domains. Hence, there exists a necessity to investigate the efficacy of utilizing domain-specific pre-trained language models and sentence encoders for processing crisis-related social media texts.

To address the above-discussed gaps in the crisis informatics literature, this study proposes *CrisisTransformers*, an ensemble of pre-trained language models and sentence encoders trained on hundreds of millions of crisis-related tweets from over 30 different crisis events, including the COVID-19 pandemic. CrisisTransformers provide valuable embeddings that enhance crisis response and management. These embeddings can be utilized in various tasks, including text classification, semantic search, clustering, and topic modelling. Advancements in these applications contribute to a more comprehensive understanding of crisis-related social media texts, thereby aiding decision-making processes and facilitating targeted interventions and communication strategies during times of crisis.

This study contributes the following to the **crisis informatics** literature:

- We provide the first set of experiments relative to domain-specific pre-training to address the following research questions:
  - How does the choice of model initialization impact pre-training in terms of loss convergence?
  - With BERTweet [10] and other strong baselines in place, can yet another domain-specific pre-trained model demonstrate superior performance in crisis-related social media text classification?
  - To what extent do domain-specific pre-trained models help generate sentence embeddings with semantic richness, in comparison to current pre-trained models and sentence encoders?

[1]Semantically rich sentence embeddings position semantically similar sentences close together in the vector space, and such sentence embeddings can be used with similarity measures such as cosine similarity.

  - Can the performance gains achieved by increasing training data size lead to substantial improvements in domain-specific sentence encoding tasks?

- We introduce CrisisTransformers, the first pre-trained language models and sentence encoders designed for processing crisis-related social media texts. The pre-training of CrisisTransformers was done on 6 NVIDIA A100 GPUs over a period of 2 months.

- Our pre-trained models outperform existing models across all 18 crisis-related datasets in classification tasks, and our best-performing sentence encoder improves the current state-of-the-art by 17.43% in sentence encoding tasks. Results confirm that CrisisTransformers can capture distinct linguistic nuances, informal language structures, and unique contextual cues present in crisis contexts.

- We publicly release CrisisTransformers, which can be used with the *Transformers* [11] library. We anticipate that these models will serve as a robust baseline for tasks involving the analysis of crisis-related social media texts.

The rest of the paper is organized as follows: Section 2 discusses related work, Section 3 details the materials and methods used in designing CrisisTransformers, Section 4 presents evaluation results and discussions, and Section 5 concludes the paper.

2. Related Work

Transformer-based models have shown remarkable success in various NLP tasks, outperforming traditional approaches and significantly advancing the state-of-the-art. The key to their performance lies in pre-training — a stage that involves training the model with unsupervised learning objectives on large-scale corpora such that the model captures contextual information and learns rich language representations. Researchers have used contextualized embeddings generated by the transformer-based models to design powerful classification/regression models and further adjusted the models to make their embeddings suitable for semantic search and clustering tasks. In this section, we review the literature associated with Encoder-only models, such

as BERT, and others, predominantly used to extract embeddings for tasks such as classification/regression, semantic search, clustering, etc.

BERT, introduced in [12], has become a ubiquitous baseline in NLP tasks. BERT uses two pre-training objectives — masked language modelling (MLM) and next sentence prediction (NSP). The MLM objective involves randomly masking specific tokens of an input sentence and training the model to predict the original masked tokens based on the context (surrounding words). Through this objective, BERT learns relationships between words and captures rich contextualized representations. Since the introduction of BERT, MLM has become a standard pre-training objective for many transformer-based models. Various improvements in training approaches and variants of MLM have been explored in subsequent research. In [13], Liu et al. proposed RoBERTa, which outperformed BERT in various downstream tasks with some changes in the pre-training process — large batch size, longer training, more training data, and removal of the NSP objective. In [14], ALBERT was introduced, which offered competitive results with reduced parameters through factorized embedding parameterization and cross-layer parameter sharing. MPNet was introduced in [15] combining MLM and permuted language modelling (PLM). In PLM [16], a sequence is randomly permuted, and the model autoregressively predicts the tokens. In [17], with XLM-RoBERTa, Conneau et al. confirmed the usefulness of pre-training multilingual language models on large-scale data containing 100 languages for cross-lingual transfer tasks. In [18], Clark et al. introduced ELECTRA, a pre-training objective where two models (generator and discriminator) are involved — the generator replaces tokens in a sequence, and the discriminator predicts which tokens are originals and which are the ones replaced by the generator. The above-discussed models were pre-trained on datasets such as *Wikipedia*, *BooksCorpus*, *OpenWebText*, *CC-News*, etc., which contain general domain texts. Researchers have also introduced domain-specific pre-trained models; we discuss some of those models next.

BERTweet [10] is a transformer-based model specifically designed for processing Twitter data and other social media texts. It leverages the BERT model configuration and incorporates RoBERTa's pre-training approach. During pre-training, it was exposed to a massive corpus containing 16 billion word tokens. BioBERT, which was introduced in [3], was pre-trained on biomedical texts, including PubMed abstracts (PubMed) and PubMed Central full-text articles (PMC), using the same architecture as BERT. Similarly, SciBERT [4] also shared the architecture of BERT but was pre-trained on a random

sample of over 1 million papers. Its pre-training corpus consisted of 18% computer science and 82% biomedical domain full-text papers. Additionally, a variant of the BERT architecture called ClinicalBERT [5] was developed by pre-training on electronic health records. This specific pre-training made ClinicalBERT suitable for processing clinical text and medical data. BERT's application has also been extended to the finance domain. FinBERT, introduced in [6], is a pre-trained model trained on an extensive financial communication corpus containing 4.9 billion tokens.

When the pre-trained models utilize either the embeddings of the *CLS* token or the mean-pooling of all tokens to generate *sentence embeddings* and subsequently undergo fine-tuning with a classification head, they produce state-of-the-art results in text classification/regression tasks. However, previous research shows that such sentence embeddings lack semanticity and are actually worse than averaging GloVe embeddings [8]. For effective semantic search and clustering tasks, it is critical to have semantically meaningful embeddings that position sentences in a vector space, such that semantically similar sentences are located closely together. Generating such sentence embeddings is an extensively researched area, and various methods have been proposed, which we discuss next.

In [19], Kiros et al. trained an encoder-decoder model to reconstruct the surrounding sentences of an encoded sequence so that the sentences that share semantic properties are mapped to similar vector representations. In [20], a siamese BiLSTM network was trained with max-pooling on the Stanford Natural Language Inference (SNLI) dataset which outperformed previous unsupervised methods [19, 21]. In [22], a transformer network was trained and unsupervised learning was extended with training on the SNLI dataset. Additionally, in [23], Yang et al. presented an unsupervised learning approach to sentence-level semantic similarity based on conversational data. Until this period, the sentence encoding approaches involved training the respective networks from scratch. After the introduction of BERT in 2018, replacing the unsupervised training part of designing sentence encoders became possible. In [8], BERT was finetuned through siamese and triplet networks on SNLI and Multi-Genre natural language inference (MultiNLI) datasets, with softmax classifier over "contradiction", "entailment", and "neutral" labels. Similarly, Gao et al. proposed SimCSE [9], a contrastive approach to finetune pre-trained models with natural language inference datasets using "contradiction" pairs as hard negatives. Following [9], Reimers and Gurevych [8] fine-tuned multiple pre-trained models using the contrastive training objec-

Figure 1: The pre-training corpus curation process.

tive on over 1 billion sentence pairs and publicly released all their models as Sentence-Transformers.

3. Materials and methods

3.1. The crisis corpus

A large-scale social media text corpus was curated for unsupervised pre-training, with Twitter serving as the primary data source. Our main objective was to create a comprehensive corpus containing texts discussing a diverse range of crisis events, such as disease outbreaks, natural disasters, terrorist attacks, conflicts, and other critical incidents. In general, as illustrated in Figure 1, the corpus underwent curation across three distinct stages: (i) consideration of an in-house dataset consisting of billions of tweets, (ii) hydration of Twitter identifiers collected from various data repositories, and (iii) utilization of Twitter's full-archive endpoint to search historical tweets. We maintained an in-house billion-scale COVID-19 tweets dataset from the onset of the COVID-19 outbreak until March 2023. The initial version of the dataset, *COV19Tweets* [24], comprised more than 2.2 billion tweets. Subsequently, we created the second version, *BillionCOV* [25], by filtering out unavailable tweets, resulting in over 1.4 billion tweets. For this study, we considered all the tweets present in BillionCOV, excluding retweets. Although BillionCOV contains COVID-19-related tweets, the COVID-19 discourse was not solely limited to discussions about the virus. Numerous other events unfolded worldwide along with the pandemic, including economic crises, natural disasters, humanitarian crises, social unrest, mental health concerns, and social issues.

Next, we collected tweet identifiers from multiple data repositories such as *CrisisNLP* [26] and *DocNow Catalog*[2]. Tweets collected from these sources needed to be hydrated to re-create the datasets locally, as Twitter's data re-distribution policy restricts sharing data other than tweet identifiers. At this stage, the corpus had texts related to more than 30 crisis events that occurred after 2014. Furthermore, to fill the temporal gap in the corpus, we utilized Twitter's full-archive endpoint to search for historical tweets created between 2006 and 2013. We applied *lang:en* condition and used the following keywords (along with their #hashtag and plural variants): crisis, disaster, earthquake, typhoon, volcano, flood, landslide, hurricane, tornado, cyclone, wildfire, famine, drought, tsunami, avalanche, epidemic, hailstorm, storm, protest, virus, war, and riot. Below are some of the crisis events covered in the corpus.

- Disease Outbreaks: COVID-19, Middle East Respiratory Syndrome, Ebola Virus Outbreak.
- Natural Disasters: Hurricanes Harvey, Irma, Florence, Dorian, Odile, Cyclone PAM, Typhoon Hagupit, California Earthquake, Pakistan Earthquake, Chile Earthquake, Nepal Earthquake, Pakistan Floods, India Floods, Iceland Volcano, Tropical Storm Imelda.
- Terrorist Attacks: Paris Attacks, Stockholm Attack, Catalonia Attacks, Peshawar School Attack.
- Protests and Activism: #J20, Tyendinaga protests.
- Shootings: Dallas Police Shooting, Las Vegas Shooting.
- Landslides: Landslides worldwide.
- Conflicts: Gaza, Palestine Conflict.
- Civil War: Fall of Aleppo.
- Missing Flight: Flight MH370.

[2]https://catalog.docnow.io/

	count
tokens	15 billion
sentences	997 million
unique tokens	36.7 million

Table 1: Descriptive statistics of the preprocessed corpus. Note: A tweet can have multiple sentences.

3.1.1. Text pre-processing

Each tweet in the corpus was pre-processed as follows: We (i) replaced URLs with "HTTPURL" token, (ii) replaced mentions (usernames) with "@USER" token (iii) decoded HTML entities to their original form (e.g., & to &), (iv) removed newline characters and replaced multiple consecutive whitespaces with a single space, (v) fixed text encoding to correct various encoding issues and improve consistency in text representation, and (vi) replaced emojis with their textual representation, as their descriptive text counterparts are meaningful. We considered only the tweets with more than ten tokens. Refer to Table 1 for the descriptive statistics of the corpus.

3.2. Unsupervised pre-training

3.2.1. Architecture and pre-training procedure

CrisisTransformers use the same architecture as $BERT_{BASE}$. In contrast to existing studies [3, 6, 5, 10], we adopted a more versatile approach to selecting a pre-training procedure for our models. Instead of starting with a specific pre-training procedure, we experimented with multiple state-of-the-art models, namely MPNet, BERTweet, BERT, RoBERTa, XLM-RoBERTa, ALBERT, and ELECTRA, on classification tasks using 18 crisis-related labelled datasets (detailed in Section 3.3.1). We observed that RoBERTa's pre-training procedure outperforms others in our domain, as RoBERTa and BERTweet emerged as the top-performing models on average. Therefore, we selected RoBERTa's pre-training procedure for training CrisisTransformers. Due to the extensive adoption of BERT and RoBERTa, we do not provide an in-depth explanation of the architecture in this paper; for more comprehensive insights, please refer to [12, 13].

3.2.2. Pre-training data

We trained a Byte-Level BPE (Byte-Pair Encoding) tokenizer using the *Tokenizers* library [11] for our domain, utilizing the pre-processed crisis corpus discussed in Section 3.1. Acknowledging the nuanced nature of social

model	intersection	unique
RoBERTa	37,338	12,927
BERTweet	15,121	48,880
BERT	7,905	21,091
XLM-RoBERTa	6,431	243,571
MPNet	5,754	24,773
ELECTRA	5,749	24,773
ALBERT	4,394	25,606

Table 2: Vocabulary similarity between existing pre-trained models and CrisisTransformers. Note: *intersection* denotes the number of tokens shared between the existing models and CrisisTransformers, while *unique* indicates the tokens exclusive to the vocabulary of the existing models.

media texts (the crisis corpus had 36 million unique tokens), we also set the vocabulary size to 64k [10]. Next, we used the trained tokenizer to tokenize the crisis corpus, thus generating sequence blocks of size 128, on which we trained the CrisisTransformers. Table 2 provides a comparative analysis of token counts in the vocabularies of established pre-trained models and CrisisTransformers. Among the existing models, RoBERTa and BERTweet share the highest similarity in vocabulary with CrisisTransformers.

3.2.3. Optimization

We pre-trained three models (as shown in Figure 2), utilizing 6 NVIDIA A100 GPUs (each with 80GB of memory). The training configurations for these models were as follows: CT-M1 (or **C**risis**T**ransformer-**M**odel**1**) was pre-trained from scratch with randomly initialized weights; CT-M2 had weights initialized with pre-trained RoBERTa's weights; and CT-M3 had weights initialized with pre-trained BERTweet's weights. CT-M1 was trained for 40 epochs, while CT-M2 and CT-M3 were trained for 20 epochs each. We used the *Transformers* library [11] to implement these models.

For optimization, we employed the *AdamW* optimizer with a peak learning rate set to 0.0004. To utilize the available GPU memory efficiently, we used a batch size of 8k with gradient accumulation steps of 16. Additionally, we set 5% of the total training steps for warming up the learning rate. All three models finished training in two months.

3.3. Fine-tuning

For fine-tuning the pre-trained models for text classification, as outlined in [10], we added a linear prediction layer to the pooled output. We imple-

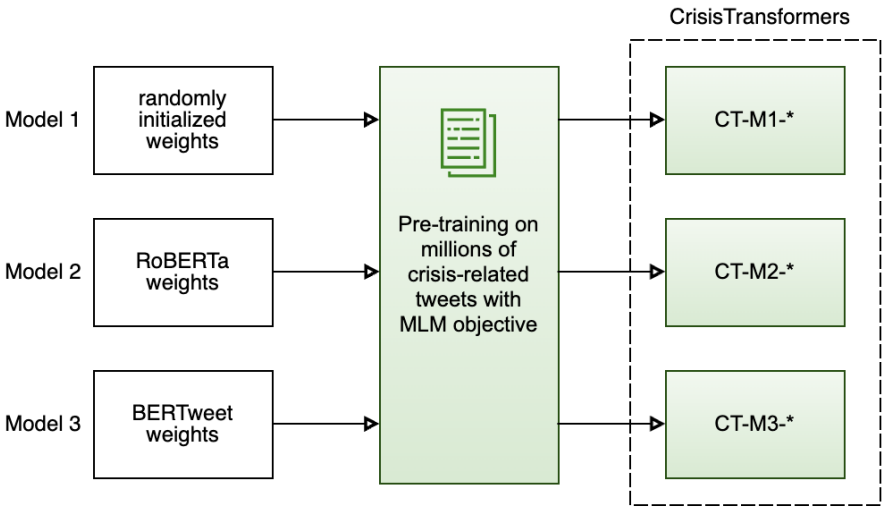

Figure 2: Pre-training of CrisisTransformers. Note: "∗" represents different checkpoints, which will be discussed later in Section 4.

mented mean pooling over the token embeddings of an input sequence while considering the attention mask. Both baselines and CrisisTransformers were fine-tuned under identical conditions. Each model was fine-tuned across 18 labelled crisis-related datasets for a maximum of 30 epochs, a batch size of 32, a learning rate of 1e-5, and AdamW as an optimizer. During each epoch, classification performance was assessed on a validation set. Early stopping was configured with a patience of 5 and a threshold of 0.0001. The final checkpoint was then used for evaluation on a test set. The fine-tuning procedure was repeated 5 times per model and dataset, with average performance scores being reported.

3.3.1. Labelled crisis-related datasets

Table 3 lists the datasets we considered to evaluate both baselines and CrisisTransformers. Evaluating the performance across such diverse datasets from the crisis informatics domain was essential to test the robustness of our proposed models. CrisisBench [27] provided the train/validation/test splits for datasets D-01 through D-06. For the remaining datasets, we implemented stratified sampling, allocating 70% for training, 10% for validation, and 20%

for testing, using scikit-learn's *train-test split*[3] with a random state of 42.

3.4. Enriching sentence encoding

By default, CrisisTransformers do not produce semantically rich embeddings, even though they were trained on a domain-specific corpus. Such pre-trained models require additional fine-tuning to learn to represent semantically similar sentences closer together within the vector space. These enhanced embeddings, capable of capturing semantic meanings, can then be effectively compared using cosine similarity. Their significance becomes particularly evident in tasks involving semantic search and clustering.

Our sentence encoders (CT-M1-*-SE, CT-M2-*-SE, and CT-M3-*-SE — where, "SE" stands for **S**entence **E**ncoder) are built upon the recent success of utilizing siamese and triplet networks on sentence pairs [20] with pre-trained transformers [8] while leveraging the idea that adding corresponding contradicting pairs as "hard negatives" alongside in-batch negatives further improves the performance [9]. Expanding upon the method introduced in [9], we adapt it to utilize domain-specific pre-trained models instead of the existing general pre-trained models like BERT and RoBERTa. We used the following contrastive learning objectives to train our sentence encoders:

- **Multiple Negative Ranking (MNR)**: This loss incorporates the (anchor, positive) pairs. Given a batch of pairs $(a_1, a_1^+), (a_2, a_2^+), \ldots, (a_n, a_n^+)$ where (a_i, a_i^+) are positive pairs and (a_i, a_j^+) for $i \neq j$ are considered negative pairs. The training objective for (a_i, a_i^+) with mini-batch N is:

$$l_i = -\log\left(\frac{e^{\text{similarity}(r_i, r_i^+)/\tau}}{\sum_{j=1}^{N} e^{\text{similarity}(r_i, r_j^+)/\tau}}\right) \tag{1}$$

  where, r_i and r_i^+ are embeddings of a_i and a_i^+ generated by our CrisisTransformers, *similarity*(r_i, r_i^+) is cosine similarity, and τ is temperature hyperparameter.

- **MNR with hard negatives**: This loss incorporates the (anchor, positive, hard negative) pairs, i.e., (a_n, a_n^+, a_n^-). The training objective in Equation 1 can be modified to:

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

Id	Dataset	Description – (# of Classes)	Samples
D-01	CrisisMMD [28]	Tweets from 7 disaster events from 2017 – (6)	10,070
D-02	CrisisLex [29]	Tweets from 26 different crisis events in 2012–13 – (6)	10,041
D-03	AIDR [30]	Tweets collected by AIDR system – (9)	5,169
D-04	ISCRAM2013 [31]	Tweets from 2 different events in 2011 – (5)	810
D-05	SWDM2013 [32]	Tweets related to Joplin tornado and Hurricane Sandy – (4)	346
D-06	CrisisNLP [26]	Tweets from 19 different disaster events in 2013–15 – (8)	10,214
D-07	Poddar et al. [33]	Tweets related to stance towards COVID-19 vaccines – (3)	3,300
D-08	SAD Stressor [34]	SMS-like sentences mentioning everyday stressors	6,850
D-09	SAD Stress [34]	Stress and non-stress SMS-like sentences – (2)	6,850
D-10	SAD COVID [34]	COVID and non-COVID SMS-like sentences – (2)	6,850
D-11	LocBERT [35]	COVID-19 tweets with origin and non-origin locations – (2)	2,800
D-12	HMC (a) [36]	Figurative versus literal health reports on Twitter – (3)	13,017
D-13	Cotfas et al. [37]	Twitter opinions regarding COVID-19 vaccination – (3)	2,393
D-14	HMC (b) [36]	Disease mentions on tweets – (10)	13,017
D-15	PHM [38]	Health mentions in social media – (4)	4,419
D-16	Klein et al. (a) [39]	Tweets about actual and potential COVID-19 patients – (3)	4,266
D-17	Klein et al. (b) [39]	Tweets about groups of potential COVID-19 positive contacts – (8)	4,266
D-18	ANTiVax [40]	Tweets on vaccine misinformation – (2)	11,518

Table 3: Labelled crisis datasets considered in this study for evaluating the performance of baselines and CrisisTransformers.

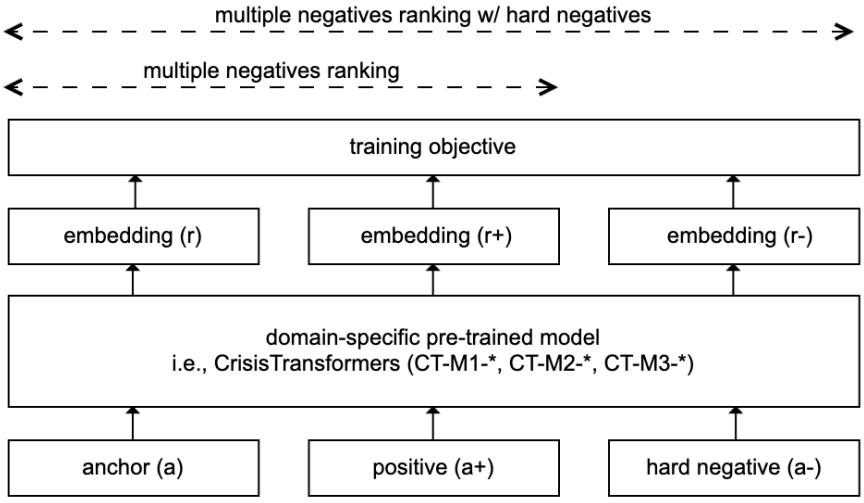

Figure 3: Training of our sentence encoders.

$$l_i = -\log\left(\frac{e^{\text{similarity}(r_i, r_i^+)/\tau}}{\sum_{j=1}^{N}(e^{\text{similarity}(r_i, r_j^+)/\tau} + e^{\text{similarity}(r_i, r_j^-)/\tau})}\right) \tag{2}$$

The MNR loss maximizes the similarity between an anchor sentence and its positive sentence while considering all other positives in a batch as negatives. In MNR with hard negatives, the similarity between an anchor sentence and its positive sentence is maximized while using its hard negative and all other positive sentences in the same batch as negatives. We include the MNR training objective in the experiments for comparison purposes, even though MNR with hard negatives has been shown to outperform it [9]. We train our sentence encoders (as shown in Figure 3) with these two objectives on (Question, Answer) pairs from GooAQ [41], (anchor, positive, hard negative) triplets from QQP[4] [42] and (anchor, entailment, contradiction) triplets from AllNLI [42, 43, 44] with a large batch size of 512 for a maximum of 20 epochs. We utilize a learning rate of 2e-05 and allocate 1% of the total training steps for warm-up.

[4]https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

3.5. Evaluation setup

3.5.1. Classification task

In line with prior research [10, 4], we evaluate baselines and CrisisTransformers for the classification task using F1-macro, which considers the precision and recall of each class and provides an overall evaluation of the models' classification performance. For each dataset, we compute the F1-macro score as follows:

$$
\begin{aligned}
P_{\text{class}_i} &= \frac{TP_{\text{class}_i}}{TP_{\text{class}_i} + FP_{\text{class}_i}} \\
R_{\text{class}_i} &= \frac{TP_{\text{class}_i}}{TP_{\text{class}_i} + FN_{\text{class}_i}} \\
F1_{\text{class}_i} &= \frac{2 \cdot P_{\text{class}_i} \cdot R_{\text{class}_i}}{P_{\text{class}_i} + R_{\text{class}_i}} \\
\text{F1-macro} &= \frac{1}{N_{\text{classes}}} \sum_{i=1}^{N_{\text{classes}}} F1_{\text{class}_i}
\end{aligned}
$$

where, TP_{class_i} is the number of true positive predictions for class i, FP_{class_i} is the number of false positive predictions for class i, FN_{class_i} is the number of false negative predictions for class i, and $N_{classes}$ is the total number of classes in the dataset.

3.5.2. Sentence encoding task

There is an absence of standard benchmark datasets in the crisis informatics domain to assess the semantic quality of the generated embeddings. In agreement with Reimers and Gurevych (2019) [8] and Gao et al. (2021) [9] that the primary objective of the embeddings is to capture and represent semantic relationships in text data, we designed an alternative task. Our evaluation method involved calculating the weighted average cosine similarity among encoded tweets within individual classes in a labelled dataset, thereby measuring the semantic consistency of tweets belonging to the same class. This approach allowed us to capture the complexities and semantics of crisis-related content, resulting in a more insightful evaluation of the sentence embeddings.

Task definition: Let N represent the total number of crisis-related tweets in a dataset and K denote the number of unique classes within the

dataset. Let E be a matrix of sentence embeddings, where each row $\mathbf{e}_i$ corresponds to the normalized embedding of the i-th tweet. Additionally, let y be a vector containing the class labels associated with each tweet.

For each unique class c_k, the class weight w_k is computed as the inverse of the count of tweets belonging to that class:

$$w_k = \frac{1}{\text{count}(c_k)}$$

These class weights are then normalized to obtain $\hat{w}_k$:

$$\hat{w}_k = \frac{w_k}{\sum_{i=1}^{K} w_i}$$

For each unique class c_k, the intra-class cosine similarity d_k is computed. For each tweet $\mathbf{e}_i$ within class c_k, the average cosine similarity to other tweets within the same class is determined:

$$d_k = \frac{1}{|\{i : y_i = c_k\}|} \sum_{i:y_i=c_k} \text{similarity}(\mathbf{e}_i, \mathbf{e}_j)$$

Here, similarity$(\mathbf{e}_i, \mathbf{e}_j)$ calculates the cosine similarity between tweet embeddings $\mathbf{e}_i$ and $\mathbf{e}_j$, where $\mathbf{e}_j$ is a tweet within the same class as $\mathbf{e}_i$.

The weighted average distance D_{avg} is computed across all classes, considering their respective normalized class weights $\hat{w}_k$:

$$D_{\text{avg}} = \sum_{k=1}^{K} \hat{w}_k \cdot d_k$$

D_{avg} quantifies the average within-class semantic similarity of crisis-related tweets while accounting for the distribution of class weights.

The cosine similarity between sentence embeddings reflects how semantically similar or related the sentences are. If the embeddings are better at capturing the semantic content of crisis-related tweets within each class, the cosine similarity values within a class would be high. A higher cosine similarity within each class indicates that the embeddings effectively represent tweets that share similar content or context related to a specific crisis-related class. In summary, the higher the value of D_{avg}, the better the performance of a sentence encoder. We considered all the datasets listed in Table 3 for this task.

4. Results and Discussion

4.1. Checkpoints and convergence

After the pre-training, we were interested in multiple checkpoints of CrisisTransformers: CT-M1-*, CT-M2-*, and CT-M3-*. CT-M1 was built from scratch and had two variants, CT-M1-BestLoss, representing the model at its lowest loss achieved during training at the 26th epoch, and CT-M1-Complete, representing the model after 40 epochs. On the other hand, CT-M2 and CT-M3 were initialized using weights from pre-trained RoBERTa and BERTweet, respectively, and were trained up to 20 epochs each. CT-M2-OneLook represents the model after 1 epoch, while CT-M2-BestLoss and CT-M2-Complete represent the model at its lowest loss and the model after 20 epochs, respectively. The same setup was applied to CT-M3 models. In total, CrisisTransformers has 8 variants based on different checkpoints of CT-M1, CT-M2, and CT-M3 models.

Figure 4 visualizes the validation loss versus epoch for each model. The graph provides insights into the impact of different initialization on the models' convergence. The loss patterns of the three models revealed distinct behaviours. CT-M1 demonstrated a gradual and consistent reduction in loss throughout the training period, suggesting steady convergence. CT-M2, on the other hand, exhibited a sharp initial drop in the loss within a few training steps, indicating rapid convergence and a smoother decline. Similarly, CT-M3 also displayed a significant initial loss drop. While CT-M3 initially shared a sharp loss drop with CT-M2, its convergence pattern aligned more with CT-M1 in the later epochs. The final loss of CT-M3 ultimately converged closer to that of CT-M1. All models seemed to plateau in their loss during the later epochs, indicating a potential convergence point. These loss patterns highlight the influence of different initializations on the time and trajectory of loss convergence; the pre-trained models seem to leverage their existing knowledge for a more efficient initial convergence than the model whose weights were randomly initialized.

4.2. Evaluations

For the classification task, we considered MPNet, BERTweet, BERT, RoBERTa, XLM-RoBERTa, ALBERT, and Electra as baselines for CrisisTransformers. As discussed in Section 3.3, we finetuned the baselines and CrisisTransformers for the classification task across 18 different crisis-related

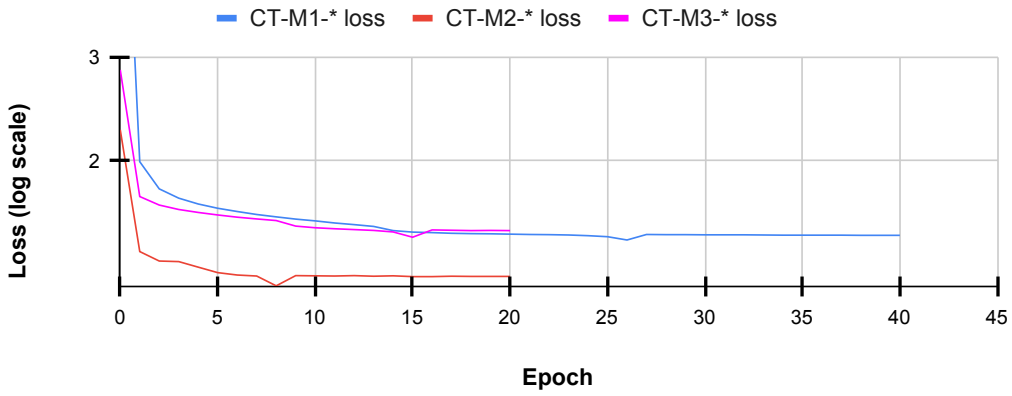

Figure 4: Validation loss versus epoch for CrisisTransformers' CT-M1-*, CT-M2-*, and CT-M3-* checkpoints, showing the impact of different initializations. The loss for CT-M1 at Epoch 0 was 9.841, and it achieved its lowest loss at the 26th epoch. For CT-M2, the loss at Epoch 0 was 2.26, and it achieved its lowest loss at the 8th epoch. Lastly, CT-M3 started with a loss of 2.856 at Epoch 0 and reached its lowest loss at the 15th epoch. The maximum value for the y-axis in this figure has been set to 3.

datasets, each identified by a unique identifier (D-01 through D-18) (refer to Table 3). Results from the experiments are summarized in Table 4.

Amongst the baselines, RoBERTa consistently outperforms other models with high F1-macro scores across several datasets. However, with the introduction of CrisisTransformers, the checkpoints of CT-M1 and CT-M2 stand out; overall, CrisisTransformers outperform the existing pre-trained models across all 18 datasets. The following models outperformed others in the respective number of datasets: CT-M1-BestLoss (in 4 datasets), CT-M1-Complete (in 3 datasets), CT-M2-OneLook (in 4 datasets), CT-M2-BestLoss (in 1 dataset), CT-M2-Complete (in 4 datasets), and CT-M3-OneLook (in 2 datasets). These results confirm the potential of CrisisTransformers for generalization and applicability in various crisis text classification tasks, which is particularly valuable for real-world applications such as disaster response, emergency communication, and crisis management.

Next, we performed sentence encoding tasks across all 18 datasets with the existing pre-trained models, CrisisTransformers, Sentence-Transformers, SimCSE, and CrisisTransformers-based sentence encoders. The results from the sentence encoding task are summarized in Tables 5–7.

The pre-trained models do not yield semantically meaningful sentence

embeddings out-of-the-box. Nevertheless, one of our objectives was to investigate how well domain-adapted models perform in generating semantically meaningful sentence embeddings. Results show that (refer to Table 5), within the existing pre-trained models, BERTweet emerged as a robust performer, consistently achieving competitive weighted average cosine similarity scores. However, CrisisTransformers, particularly the CT-M3 variants, invariably achieved the highest or second-highest scores regardless of the specific configuration (OneLook, BestLoss, or Complete). The performance of BERTweet and CT-M3 variants can be attributed to BERTweet's pre-training on an extensive corpus of tweets. The results further indicate that the performance of the pre-trained RoBERTa is subpar. Consequently, the CT-M2 variants do not notably enhance performance. In contrast, the CT-M1 variants demonstrate a significant performance advantage over the CT-M2 variants. These findings suggest that further pre-training a domain-specific model on a sub-domain corpus (where, "tweets" reflect "domain" and "crisis-related tweets" indicate "sub-domain") leads to improved performance in generating better sentence embeddings.

Furthermore, we trained CrisisTransformers using siamese and triplet networks with MNR and MNR with hard negatives training objectives, as discussed in Section 3.4, to create sentence encoders specifically designed for crisis-related social media texts. We used GooAQ (Question, Answer) pairs for MNR, and QQP (anchor, positive, hard negative) triplets for MNR with hard negatives. As baselines for our sentence encoders, we utilized Sentence-Transformers and SimCSE. We considered the "all-mpnet-base-v2" model, which is the highest-performing model in Sentence-Transformers, and the "sup-simcse-roberta-base" model, a high-performing base architecture model for SimCSE. We used only the first 10k pairs from GooAQ and QQP, for both training objectives. We explored different sample sizes and ultimately found that using 10k pairs struck a balance between model performance and having fewer training samples. This is in contrast to our baselines, where Sentence-Transformers was trained on over 1 billion samples, and SimCSE was trained on 314k samples.

Table 6 and Table 7 summarize the performance of the baselines and our sentence encoders in terms of the weighted average cosine similarity, and Table 8 reports the overall performance. Across all 18 datasets, our sentence encoders outperform both Sentence-Transformers and SimCSE. Notably, CT-M1-Complete-SE (MNR) and CT-M2-Complete-SE (hard negatives) each achieved the best performances across 4 datasets, and CT-M1-

BestLoss-SE (MNR) and CT-M2-BestLoss-SE (hard negatives) each in 3 datasets. Overall, CT-M1 variants performed better in 11 datasets, CT-M2 variants in 6 datasets, and CT-M3 in 1 dataset. Considering training objectives, models trained with hard negatives achieved the highest scores across 11 datasets. CT-M1-Complete-SE (hard negatives), although trained on 10k samples, achieved an average score of 0.7140, surpassing the current state-of-the-art by 12% while outperforming Sentence-Transformers' average score of 0.6374. These results highlight the adaptability and effectiveness of CrisisTransformers-based sentence encoders in capturing semantic similarity within sentences, particularly in crisis-related contexts. This reinforces the idea that tailoring models to specific domains, like crisis situations, can yield significant improvements over more general-purpose models in sentence encoding tasks, even when trained with less data. Among the baselines, Sentence-Transformers performed better compared to SimCSE across all datasets. In fact, our CT-M3 variants (avg. scores ranging from 0.2663 to 0.2792) outperformed SimCSE (avg. score of 0.1765). The noticeable performance advantage of Sentence-Transformers over SimCSE can be attributed to the comprehensive training of its "all-mpnet-base-v2" model, which involved training on more than 1 billion sentence pairs/triplets. This extensive training likely provided the model with a broader and richer understanding of general language nuances, thus contributing to its superior performance.

Motivated to study the effect of training samples, we re-trained CT-M1-Complete-SE (hard negatives) while increasing the training samples from 10k to 102k samples (complete QQP) and further augmented the AllNLI dataset to create a training size of 378k. After this re-training, we observed an improvement of approx. 3.56% with complete QQP and approx. 4.83% with QQP+AllNLI. Overall, our best-performing sentence encoder improved the current state-of-the-art by around 17.43%. This observation sets the stage for potential enhancements to our sentence encoder. Going forward, our future objectives include training our sentence encoders on a scale similar to Sentence-Transformers for an even more substantial improvement.

model	D-01	D-02	D-03	D-04	D-05	D-06	D-07	D-08	D-09
MPNet	0.6559	0.7571	0.4286	0.7359	0.57	0.7776	0.581	0.6934	0.6758
BERTweet	0.6597	0.7569	0.5735	0.6979	0.6574	0.7871	0.589	0.7147	0.694
BERT	0.6728	0.7297	0.5956	0.6089	0.6496	0.7782	0.5197	0.7039	0.677
RoBERTa	0.6891	0.76	0.624	0.765	0.7683	0.7919	0.5809	0.7171	0.7122
XLM-RoBERTa	0.6419	0.7505	0.5908	0.6321	0.3849	0.7918	0.5187	0.7232	0.6702
ALBERT	0.6548	0.7281	0.5571	0.5589	0.4484	0.7648	0.4963	0.6982	0.6989
ELECTRA	0.6427	0.7409	0.5766	0.5422	0.2685	0.7874	0.5699	0.7183	0.7044
CT-M1-BestLoss	0.6555	0.7539	0.6184	0.673	**0.851**	**0.792**	**0.7118**	0.7121	0.7023
CT-M1-Complete	0.6567	0.7613	0.6291	0.687	0.8159	0.7875	0.7036	0.7091	0.698
CT-M2-OneLook	**0.6916**	0.757	**0.665**	**0.7744**	0.8274	0.7862	0.6504	0.725	0.7046
CT-M2-BestLoss	0.6646	0.764	0.6637	0.7669	0.7875	0.786	0.6721	0.7207	0.6969
CT-M2-Complete	0.6606	**0.7656**	0.6569	0.7453	0.8343	0.7796	0.6621	0.7015	0.6889
CT-M3-OneLook	0.6494	0.7592	0.5383	0.7021	0.8048	0.7779	0.6586	**0.729**	**0.7444**
CT-M3-BestLoss	0.6546	0.7585	0.5555	0.7165	0.5725	0.7874	0.6729	0.7174	0.72
CT-M3-Complete	0.6547	0.759	0.5725	0.7011	0.6806	0.7898	0.6814	0.7156	0.7061
model	**D-10**	**D-11**	**D-12**	**D-13**	**D-14**	**D-15**	**D-16**	**D-17**	**D-18**
MPNet	0.9208	0.759	0.8883	0.8119	0.9905	0.8029	0.804	0.5259	0.9829
BERTweet	0.9358	0.7727	0.9	0.8562	0.9933	0.8209	0.8109	0.5274	0.983
BERT	0.9001	0.723	0.8745	0.7595	0.9899	0.8106	0.7662	0.594	0.9748
RoBERTa	0.9125	0.7665	0.8904	0.8338	0.9927	0.831	0.7998	0.6182	0.9837
XLM-RoBERTa	0.9166	0.7703	0.8835	0.7975	0.993	0.8101	0.7901	0.528	0.9774
ALBERT	0.8759	0.7623	0.875	0.7625	0.9917	0.805	0.8011	0.5709	0.976
ELECTRA	0.9005	0.782	0.8878	0.8269	0.9913	0.8119	0.8008	0.5027	0.9813
CT-M1-BestLoss	0.9336	0.7855	0.8988	**0.8654**	0.9929	0.826	0.8537	0.5663	0.9885
CT-M1-Complete	0.94	0.7818	**0.9043**	0.8641	0.9916	0.8343	**0.855**	0.5572	**0.9884**
CT-M2-OneLook	0.9337	0.7775	0.8988	0.8614	0.9928	**0.8426**	0.8407	0.708	0.9851
CT-M2-BestLoss	0.9393	0.7875	0.8953	0.849	**0.994**	0.8265	0.8244	0.6304	0.9861
CT-M2-Complete	**0.9476**	**0.7903**	0.8978	0.8491	0.993	0.84	0.8207	**0.725**	0.9881
CT-M3-OneLook	0.9386	0.7861	0.8947	0.8268	0.9938	0.8234	0.8319	0.6025	0.9866
CT-M3-BestLoss	0.9439	0.7829	0.8968	0.8595	0.9938	0.8388	0.8328	0.6235	0.9865
CT-M3-Complete	0.9398	0.7785	0.8961	0.8509	0.9933	0.8103	0.8338	0.647	0.9494

Table 4: Performance of the existing pre-trained models and CrisisTransformers (CT-*) on classification task across 18 crisis datasets (D-01 through D-18), with average F1-macro being reported. For the corresponding dataset names of each dataset identifier, please refer to Table 3. The best scores are shown in **bold**.

model	D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
MPNet	0.0709	0.0777	0.0783	0.0795	0.073	0.0739	0.0618	0.0938	0.1719
BERTweet	0.2532	0.2508	0.2687	0.2349	0.218	0.2382	0.2255	0.2465	0.299
BERT	0.0897	0.1182	0.1108	0.121	0.1079	0.1076	0.1171	0.1904	0.2429
RoBERTa	0.0235	0.0273	0.0272	0.0272	0.0252	0.0254	0.0276	0.037	0.0502
XLM-RoBERTa	0.0026	0.0027	0.0028	0.0026	0.0024	0.0025	0.0024	0.0041	0.0074
ALBERT	0.1129	0.1249	0.1262	0.1286	0.1088	0.1183	0.1225	0.1854	0.272
ELECTRA	0.0646	0.0713	0.0805	0.0713	0.063	0.0665	0.0554	0.0898	0.1702
CT-M1-BestLoss	0.1129	0.1238	0.1216	0.1117	0.1094	0.1243	0.1303	0.1137	0.1407
CT-M1-Complete	0.1078	0.1186	0.1164	0.1064	0.1052	0.1199	0.127	0.1111	0.1368
CT-M2-OneLook	0.0309	0.036	0.034	0.0346	0.0328	0.032	0.0378	0.0456	0.0625
CT-M2-BestLoss	0.0527	0.0581	0.0548	0.0561	0.0515	0.052	0.0676	0.0766	0.1019
CT-M2-Complete	0.0541	0.0584	0.0552	0.0564	0.0521	0.0529	0.0713	0.0731	0.097
CT-M3-OneLook	0.2712	**0.2962**	0.2636	**0.2694**	**0.2578**	0.2648	**0.2921**	**0.273**	**0.3179**
CT-M3-BestLoss	**0.2758**	0.2935	**0.2697**	0.2579	0.2468	**0.2673**	0.2867	0.253	0.3008
CT-M3-Complete	0.2732	0.2853	0.2679	0.2492	0.2404	0.2661	0.2858	0.2432	0.2874

model	D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
MPNet	0.1119	0.053	0.0961	0.0629	0.0962	0.0813	0.0654	0.054	0.0665
BERTweet	0.2475	0.1997	0.3069	0.2261	0.3039	0.2776	0.2283	0.1949	0.2452
BERT	0.1839	0.102	0.1684	0.1076	0.162	0.1587	0.1423	0.1154	0.1309
RoBERTa	0.0354	0.0246	0.0372	0.0262	0.0378	0.0331	0.0292	0.0239	0.0302
XLM-RoBERTa	0.0042	0.0021	0.004	0.0026	0.004	0.0032	0.0033	0.0023	0.003
ALBERT	0.1933	0.1109	0.1666	0.1246	0.1671	0.1516	0.1291	0.1069	0.1366
ELECTRA	0.1134	0.0439	0.0922	0.0596	0.0916	0.0796	0.0665	0.0416	0.0648
CT-M1-BestLoss	0.1188	0.1168	0.1454	0.1279	0.1451	0.1324	0.1173	0.0986	0.136
CT-M1-Complete	0.1161	0.1135	0.1409	0.1248	0.1403	0.1285	0.115	0.0977	0.1328
CT-M2-OneLook	0.0445	0.0317	0.0505	0.0353	0.0509	0.0444	0.0372	0.0314	0.0407
CT-M2-BestLoss	0.078	0.0503	0.0791	0.0631	0.0786	0.0697	0.0627	0.0587	0.0724
CT-M2-Complete	0.0744	0.0528	0.0791	0.0651	0.0788	0.0691	0.0641	0.0601	0.0751
CT-M3-OneLook	**0.2831**	**0.2644**	**0.3284**	**0.2814**	**0.3194**	**0.3028**	**0.2418**	**0.2107**	**0.2886**
CT-M3-BestLoss	0.2586	0.2537	0.3224	0.2782	0.3162	0.2938	0.2315	0.1996	0.2829
CT-M3-Complete	0.25	0.2519	0.3122	0.2795	0.3071	0.2844	0.2283	0.1965	0.2856

Table 5: Performance of the existing pre-trained models and CrisisTransformers on sentence encoding task across 18 crisis datasets (D-01 through D-18), with weighted average cosine similarity being reported. The best scores are shown in **bold**, and the second-best scores are underlined. Note that results reported in this table are intended for comparative purposes only; embeddings generated by pre-trained models out-of-the-box do not produce semantically meaningful sentence embeddings.

model	D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
Sentence-Transformers	0.6103	0.6809	0.5407	0.5632	0.487	0.5698	0.6019	0.7528	0.8749
SimCSE	0.18	0.18	0.15	0.17	0.16	0.17	0.17	0.19	0.23
CT-M1-BestLoss-SE_{MNR}	0.6201	**0.7647**	0.5936	0.59	**0.6276**	0.6344	0.8147	0.7117	0.8438
CT-M1-Complete-SE_{MNR}	0.6278	<u>0.7628</u>	0.5964	0.5904	<u>0.6229</u>	0.6382	<u>0.8212</u>	0.7156	0.8469
CT-M2-OneLook-SE_{MNR}	0.5535	0.7057	0.5418	0.5652	0.5297	0.5594	0.7277	0.7149	0.8667
CT-M2-BestLoss-SE_{MNR}	0.587	0.735	0.535	0.5968	0.5542	0.5794	0.8034	0.7136	0.8705
CT-M2-Complete-SE_{MNR}	0.5834	0.7366	0.5391	0.5955	0.5573	0.5834	0.8002	0.71	0.8721
CT-M3-OneLook-SE_{MNR}	0.5672	0.7155	0.5484	0.5756	0.5459	0.5709	0.7593	0.701	0.8275
CT-M3-BestLoss-SE_{MNR}	0.5971	0.7265	0.5528	0.5864	0.5378	0.5922	0.778	0.6924	0.8215
CT-M3-Complete-SE_{MNR}	0.6002	0.7262	0.5556	0.582	0.5382	0.5951	0.785	0.6937	0.825
CT-M1-BestLoss-SE_{MNR} w/ hard negatives	**0.6465**	0.7338	<u>0.6059</u>	0.615	0.6016	**0.6649**	0.7874	0.7657	0.871
CT-M1-Complete-SE_{MNR} w/ hard negatives	<u>0.6432</u>	0.7385	**0.6097**	0.6151	0.6053	<u>0.6626</u>	0.7912	0.7685	<u>0.8735</u>
CT-M2-OneLook-SE_{MNR} w/ hard negatives	0.5702	0.6753	0.5536	0.5924	0.5465	0.5709	0.7063	0.7554	0.8733
CT-M2-BestLoss-SE_{MNR} w/ hard negatives	0.5855	0.7145	0.5436	0.6123	0.6163	0.5903	0.8158	**0.7784**	**0.8833**
CT-M2-Complete-SE_{MNR} w/ hard negatives	0.5972	0.7208	0.5466	0.6116	0.6118	0.6049	**0.8261**	<u>0.7769</u>	**0.8883**
CT-M3-OneLook-SE_{MNR} w/ hard negatives	0.605	0.7022	0.5583	0.6147	0.5674	0.595	0.7504	0.7301	0.828
CT-M3-BestLoss-SE_{MNR} w/ hard negatives	0.6307	0.715	0.5687	**0.6199**	0.5529	0.6151	0.7652	0.7334	0.8289
CT-M3-Complete-SE_{MNR} w/ hard negatives	0.6314	0.7144	0.5684	<u>0.6171</u>	0.5518	0.6161	0.7682	0.7307	0.8274

Table 6: (Part 1/2) Performance of baselines and our sentence encoders in the sentence encoding task. The best scores are shown in **bold**, and the second-best scores are <u>underlined</u>.

model	D-10	D-11	D-12	D-13	D-14	D-15	D-16	D-17	D-18
Sentence-Transformers	0.6674	0.7297	0.8211	0.4939	0.7259	0.7579	0.532	0.5192	0.545
SimCSE	0.19	0.18	0.2	0.15	0.19	0.19	0.1683	0.15	0.16
CT-M1-BestLoss-SE_{MNR}	0.6878	<u>0.8553</u>	<u>0.8426</u>	0.7094	**0.7609**	0.7624	<u>0.6788</u>	<u>0.5874</u>	0.727
CT-M1-Complete-SE_{MNR}	0.6904	**0.8584**	**0.8439**	0.7146	<u>0.7599</u>	0.7567	**0.6814**	**0.5918**	0.728
CT-M2-OneLook-SE_{MNR}	0.685	0.7972	0.823	0.6105	0.7357	0.7403	0.6182	0.5505	0.6392
CT-M2-BestLoss-SE_{MNR}	0.7075	0.838	0.8375	0.7005	0.7551	0.7424	0.6047	0.5555	0.6978
CT-M2-Complete-SE_{MNR}	0.7035	0.8425	0.8383	0.6952	0.7597	0.7432	0.6124	0.5727	0.6824
CT-M3-OneLook-SE_{MNR}	0.6633	0.8135	0.8097	0.6797	0.7247	0.7391	0.6155	0.536	0.6602
CT-M3-BestLoss-SE_{MNR}	0.6659	0.8137	0.8107	0.6831	0.7346	0.7421	0.6508	0.5655	0.6778
CT-M3-Complete-SE_{MNR}	0.6644	0.8175	0.8134	0.6907	0.7347	0.7464	0.652	0.5657	0.6857
CT-M1-BestLoss-SE_{MNR} w/ hard negatives	<u>0.7424</u>	0.7961	0.8232	0.7455	0.744	<u>0.7867</u>	0.6526	0.5346	0.727
CT-M1-Complete-SE_{MNR} w/ hard negatives	0.7417	0.7912	0.8205	0.7496	0.7441	**0.7868**	0.6547	0.531	0.7261
CT-M2-OneLook-SE_{MNR} w/ hard negatives	0.7205	0.7013	0.7833	0.626	0.7207	0.7593	0.5885	0.5022	0.6529
CT-M2-BestLoss-SE_{MNR} w/ hard negatives	**0.7478**	0.7925	0.8013	<u>0.7532</u>	0.7453	0.7759	0.6333	0.5363	<u>0.7323</u>
CT-M2-Complete-SE_{MNR} w/ hard negatives	0.7399	0.7851	0.8094	**0.7632**	0.7564	0.7815	0.6416	0.5417	**0.743**
CT-M3-OneLook-SE_{MNR} w/ hard negatives	0.704	0.7404	0.7825	0.695	0.7138	0.7482	0.6241	0.524	0.6728
CT-M3-BestLoss-SE_{MNR} w/ hard negatives	0.7092	0.7419	0.7885	0.7032	0.7235	0.7641	0.6355	0.5378	0.6883
CT-M3-Complete-SE_{MNR} w/ hard negatives	0.7071	0.7442	0.7894	0.7049	0.7241	0.7635	0.6361	0.543	0.6903

Table 7: (Part 2/2) Performance of baselines and our sentence encoders in the sentence encoding task. The best scores are shown in **bold**, and the second-best scores are <u>underlined</u>.

sentence encoder	**avg. score**
Sentence-Transformers	0.6374
SimCSE	0.1765
10k training samples of GooAQ	
– CT-M1-BestLoss-SE$_{\text{MNR}}$	0.7117
– CT-M1-Complete-SE$_{\text{MNR}}$	0.7137
– CT-M2-OneLook-SE$_{\text{MNR}}$	0.6646
– CT-M2-BestLoss-SE$_{\text{MNR}}$	0.6896
– CT-M2-Complete-SE$_{\text{MNR}}$	0.6904
– CT-M3-OneLook-SE$_{\text{MNR}}$	0.6696
– CT-M3-BestLoss-SE$_{\text{MNR}}$	0.6793
– CT-M3-Complete-SE$_{\text{MNR}}$	0.6817
10k training samples of QQP	
– CT-M1-BestLoss-SE$_{\text{MNR w/ hard negatives}}$	0.7135
– CT-M1-Complete-SE$_{\text{MNR w/ hard negatives}}$	0.7140
– CT-M2-OneLook-SE$_{\text{MNR w/ hard negatives}}$	0.661
– CT-M2-BestLoss-SE$_{\text{MNR w/ hard negatives}}$	0.7032
– CT-M2-Complete-SE$_{\text{MNR w/ hard negatives}}$	0.7081
– CT-M3-OneLook-SE$_{\text{MNR w/ hard negatives}}$	0.6753
– CT-M3-BestLoss-SE$_{\text{MNR w/ hard negatives}}$	0.6845
– CT-M3-Complete-SE$_{\text{MNR w/ hard negatives}}$	0.6848
Complete QQP (102k training samples)	
– CT-M1-Complete-SE$_{\text{MNR w/ hard negatives}}$	0.7394
Complete QQP and AllNLI (378k training samples)	
– CT-M1-Complete-SE$_{\text{MNR w/ hard negatives}}$	**0.7485**

Table 8: Overall performance of the evaluated sentence encoders across 18 datasets.

5. Conclusion

In this study, we introduced CrisisTransformers, an ensemble of pre-trained language models and sentence encoders designed for processing crisis-related social media texts. The pre-trained models were trained on a large-scale corpus of over 15 billion word tokens sourced from tweets associated with more than 30 crisis events that occurred between 2006 and 2023. Additionally, we fine-tuned the pre-trained models using siamese and triplet networks to create sentence encoders. Existing models and CrisisTransformers were evaluated on 18 crisis-specific datasets for classification and sentence encoding tasks. Our pre-trained models outperform strong baselines across all 18 datasets in classification tasks, and our best-performing sentence encoder improves the state-of-the-art by 17.43% in sentence encoding tasks. We

publicly release CrisisTransformers, which include 8 variants of pre-trained models and the best-performing sentence encoder, hoping that they will serve as a robust baseline for tasks that involve processing crisis-related social media texts.

CrisisTransformers offers checkpoints of models trained from scratch (CT-M1) and those initialized with RoBERTa's weights (CT-M2) and BERTweet's weights (CT-M3). During experimentations, we observed that pre-trained models (CT-M2 and CT-M3), which undergo further pre-training, leverage existing knowledge for efficient initial convergence, unlike randomly initialized CT-M1. CT-M2 and CT-M3 exhibited rapid initial drops in loss; CT-M3 later aligned with CT-M1 in terms of final loss. All models plateaued, implying convergence. In classification, CT-M1 performed best on 7 datasets, CT-M2 on 9, and CT-M3 on 2. Regarding sentence encoding, CT-M1 outperformed in 11 datasets, CT-M2 on 6, and CT-M3 on 1. Considering the training objectives, models trained with hard negatives achieved the highest scores across 11 datasets, which remains in line with what has been reported in the literature. We noticed that the CT-M1 at the lowest loss utilizing only 10k training samples with the MNR with hard negatives training objective outperformed the state-of-the-art Sentence-Transformers (trained on 1 billion samples) by a significant margin of 12%. By increasing the training samples to 378k using the QQP+AllNLI datasets, the performance improved further to 17.43%. This observation confirmed that domain-specific pre-trained models demonstrate significant improvements over general-purpose models in sentence encoding tasks. Going forward, our future objectives include training the sentence encoders on a scale similar to Sentence-Transformers. Also, the proposed models process only English-language tweets. As a future task, we aim to release their multi-lingual versions.

Disclaimer

The training corpus used by CrisisTransformers had a significant volume of unfiltered tweets, which inherently carry non-neutral content. As a result, both the pre-trained models and their finetuned versions are susceptible to biased predictions.

Acknowledgements

This study is supported by the Melbourne Research Scholarship from the University of Melbourne, Australia. This research was undertaken using the

LIEF HPC-GPGPU Facility hosted at the University of Melbourne, which was established with the assistance of LIEF Grant LE170100200. The cloud infrastructure required to maintain *COV19Tweets* over the last three years was provided by DigitalOcean. We appreciate the insights provided by Dat Quoc Nguyen (BERTweet's co-author) during the pre-training phase of CrisisTransformers.

CRediT authorship contribution statement

Rabindra Lamsal performed Conceptualization, Data curation, Methodology, Software, Visualization, Writing–first draft. **Maria Rodriguez Read** and **Shanika Karunasekera** performed Conceptualization, Supervision, Writing–Review & Editing.

Declaration of competing interest

All authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Imran, C. Castillo, F. Diaz, S. Vieweg, Processing social media messages in mass emergency: A survey, ACM Computing Surveys (CSUR) 47 (4) (2015) 1–38.

[2] R. Lamsal, A. Harwood, M. R. Read, Socially enhanced situation awareness from microblogs using artificial intelligence: A survey, ACM Computing Surveys 55 (4) (2022) 1–38.

[3] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (4) (2020) 1234–1240.

[4] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620.

[5] K. Huang, J. Altosaar, R. Ranganath, Clinicalbert: Modeling clinical notes and predicting hospital readmission, arXiv preprint arXiv:1904.05342 (2019).

[6] Y. Yang, M. C. S. Uy, A. Huang, Finbert: A pretrained language model for financial communications, arXiv preprint arXiv:2006.08097 (2020).

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[8] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992.

[9] T. Gao, X. Yao, D. Chen, SimCSE: Simple contrastive learning of sentence embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 6894–6910.

[10] D. Q. Nguyen, T. Vu, A. Tuan Nguyen, BERTweet: A pre-trained language model for English tweets, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 9–14.

[11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45.

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[14] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942 (2019).

[15] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, Advances in Neural Information Processing Systems 33 (2020) 16857–16867.

[16] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Advances in neural information processing systems 32 (2019).

[17] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8440–8451.

[18] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, arXiv preprint arXiv:2003.10555 (2020).

[19] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, S. Fidler, Skip-thought vectors, Advances in neural information processing systems 28 (2015).

[20] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 670–680.

[21] F. Hill, K. Cho, A. Korhonen, Learning distributed representations of sentences from unlabelled data, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 1367–1377.

[22] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al., Universal sentence encoder, arXiv preprint arXiv:1803.11175 (2018).

[23] Y. Yang, S. Yuan, D. Cer, S.-y. Kong, N. Constant, P. Pilar, H. Ge, Y.-H. Sung, B. Strope, R. Kurzweil, Learning semantic textual similarity from conversations, in: Proceedings of the Third Workshop on Representation Learning for NLP, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 164–174.

[24] R. Lamsal, Design and analysis of a large-scale covid-19 tweets dataset, applied intelligence 51 (2021) 2790–2804.

[25] R. Lamsal, M. R. Read, S. Karunasekera, Billioncov: An enriched billion-scale collection of covid-19 tweets for efficient hydration, Data in Brief 48 (2023) 109229.

[26] M. Imran, P. Mitra, C. Castillo, Twitter as a lifeline: Human-annotated Twitter corpora for NLP of crisis-related messages, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), European Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp. 1638–1643.

[27] F. Alam, H. Sajjad, M. Imran, F. Ofli, Crisisbench: Benchmarking crisis-related social media datasets for humanitarian information processing, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 15, 2021, pp. 923–932.

[28] F. Alam, F. Ofli, M. Imran, Crisismmd: Multimodal twitter datasets from natural disasters, in: Proceedings of the international AAAI conference on web and social media, Vol. 12, 2018.

[29] A. Olteanu, C. Castillo, F. Diaz, S. Vieweg, Crisislex: A lexicon for collecting and filtering microblogged communications in crises, in: Pro-

ceedings of the international AAAI conference on web and social media, Vol. 8, 2014, pp. 376–385.

[30] M. Imran, C. Castillo, J. Lucas, P. Meier, S. Vieweg, Aidr: Artificial intelligence for disaster response, in: Proceedings of the 23rd international conference on world wide web, 2014, pp. 159–162.

[31] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, P. Meier, Extracting information nuggets from disaster-related messages in social media., Iscram 201 (3) (2013) 791–801.

[32] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, P. Meier, Practical extraction of disaster-relevant information from social media, in: Proceedings of the 22nd international conference on world wide web, 2013, pp. 1021–1024.

[33] S. Poddar, M. Mondal, J. Misra, N. Ganguly, S. Ghosh, Winds of change: Impact of covid-19 on vaccine-related opinions of twitter users, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 16, 2022, pp. 782–793.

[34] M. L. Mauriello, T. Lincoln, G. Hon, D. Simon, D. Jurafsky, P. Paredes, Sad: A stress annotated dataset for recognizing everyday stressors in sms-like conversational systems, in: Extended abstracts of the 2021 CHI conference on human factors in computing systems, 2021, pp. 1–7.

[35] R. Lamsal, A. Harwood, M. R. Read, Where did you tweet from? inferring the origin locations of tweets based on contextual information, in: 2022 IEEE International Conference on Big Data (Big Data), IEEE, 2022, pp. 3935–3944.

[36] R. Biddle, A. Joshi, S. Liu, C. Paris, G. Xu, Leveraging sentiment distributions to distinguish figurative from literal health reports on twitter, in: Proceedings of the web conference 2020, 2020, pp. 1217–1227.

[37] L.-A. Cotfas, C. Delcea, I. Roxin, C. Ioanăş, D. S. Gherai, F. Tajariol, The longest month: analyzing covid-19 vaccination opinions dynamics from tweets in the month following the first vaccine announcement, Ieee Access 9 (2021) 33203–33223.

[38] P. Karisani, E. Agichtein, Did you really just have a heart attack? towards robust detection of personal health mentions in social media, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 137–146.

[39] A. Z. Klein, A. Magge, K. O'Connor, J. I. Flores Amaro, D. Weissenbacher, G. Gonzalez Hernandez, Toward using twitter for tracking covid-19: a natural language processing pipeline and exploratory data set, Journal of medical Internet research 23 (1) (2021) e25314.

[40] K. Hayawi, S. Shahriar, M. A. Serhani, I. Taleb, S. S. Mathew, Anti-vax: a novel twitter dataset for covid-19 vaccine misinformation detection, Public health 203 (2022) 23–30.

[41] D. Khashabi, A. Ng, T. Khot, A. Sabharwal, H. Hajishirzi, C. Callison-Burch, GooAQ: Open question answering with diverse answer types, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 421–433.

[42] Datasets at hugging face: Training data for text embedding models, `https://huggingface.co/datasets/sentence-transformers/embedding-training-data`.

[43] S. R. Bowman, G. Angeli, C. Potts, C. D. Manning, A large annotated corpus for learning natural language inference, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 632–642.

[44] A. Williams, N. Nangia, S. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, 2018, pp. 1112–1122.