

Adapting Transformers for Detecting Emergency Events on Social Media

Emanuela Boros¹^a, Gaël Lejeune²^b, Mickaël Coustaty¹^c and Antoine Doucet¹^d

¹University of La Rochelle, L3i, F-17000, La Rochelle, France

²Sorbonne University, F-75006, Paris, France

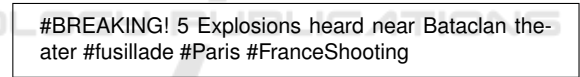
Keywords: Event Detection, Emergency Event Detection, Social Media, Language Models, Transformers.

Abstract: Detecting emergency events on social media could facilitate disaster monitoring by categorizing and prioritizing tweets in catastrophic situations to assist emergency service operators. However, the high noise levels in tweets, combined with the limited publicly available datasets have rendered the task difficult. In this paper, we propose an enhanced multitask Transformer-based model that highlights the importance of entities, event descriptions, and hashtags in tweets. This approach includes a Transformer encoder with several layers over the sequential token representation provided by a pre-trained language model that acts as a task adapter for detecting emergency events in noisy data. We conduct an evaluation on the Text REtrieval Conference (TREC) 2021 Incident Streams (IS) track dataset, and we conclude that our proposed approach brought considerable improvements to emergency social media classification.

1 INTRODUCTION

Detecting major events regarding natural disasters such as hurricanes, tsunamis, tornadoes, earthquakes, and floods, that are shared and communicated on social media streams of uninterrupted but noisy content is not trivial. In this context, the TREC¹ Incident Streams campaign (TREC-IS) (McCreadie et al., 2019; McCreadie et al., 2020; Buntain et al., 2020) aims at producing a series of curated feeds containing social media posts (Twitter), where each feed corresponds to a particular type of information request, aid request, or report containing a particular type of information. The TREC-IS track consists in producing two outputs for crisis-related social media content: classifying tweets by *information type* and ranking tweets by their criticality or *priority*. A tweet can have multiple high-level *information types* from an ontology² that may be of interest to public safety personnel and can have one of the four *priority types*: critical, high, medium, and low. A set of six *information*

types (*GoodsServices*, *SearchAndRescue*, *MovePeople*, *EmergingThreats*, *NewSubEvent*, *ServiceAvailable*) are considered *actionable* (e.g., *GoodsServices* asks for a service to be provided).




```
#BREAKING! 5 Explosions heard near Bataclan theater #fusillade #Paris #FranceShooting
```


Figure 1: High priority tweet [ThirdPartyObservation, EmergingThreats, News] that represents a *bombing* during 2015 Paris attacks.


For example, Figure 1 presents a tweet regarding the series of coordinated terrorist attacks that occurred on Friday, 13 November 2015 in Paris, France³. This tweet covers an event of type *bombing* of *critical* priority, with three *information types* (*ThirdPartyObservation*, *EmergingThreats*, *News*). Thus, detecting emergency events comprises a multilabel *information type* and a multiclass *priority* classification.


Most of the TREC-IS approaches are based on bag of word representations and classical machine learning techniques such as support vector machine (SVM), logistic regression, or random forests (Miyazaki et al., 2018; Chy et al., 2018; García-Cumbreras et al., 2018; Choi et al., 2018). Although these methods tended to overestimate, they were more

³https://en.wikipedia.org/wiki/November_2015_Paris_attacks

^a <https://orcid.org/0000-0001-6299-9452>

^b <https://orcid.org/0000-0002-4795-2362>

^c <https://orcid.org/0000-0002-0123-439X>

^d <https://orcid.org/0000-0001-6160-3356>

¹Text REtrieval Conference <http://trec.nist.gov>

²The ontology can be found at http://dcs.gla.ac.uk/~richardm/TREC_IS/2019/ITR-H.types.v3.json contains 25 high-level information types.

accurate at estimating information criticality. Other types of text representations were also leveraged by, for example, converting tweets into a form of word or character sequence embedding (e.g., ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), etc.) (Wang and Lillis, 2021; Wang et al., 2021). However, the traditional machine learning remained competitive, and moreover, the most effective systems to identify *actionable* content (Dusart et al., 2019; Mishra and Pal, 2019; Miyazaki et al., 2019).

In this paper, we propose a Transformer-based model that relies on a pre-trained and fine-tuned language model encoder and a task adapter based on a Transformer encoder with several Transformer layers. We train the model in a multitask manner to perform both multilabel *information type* and multiclass *priority* level classification. Furthermore, we augment the input tweets in order that our model becomes mode task-specific by taking advantage of the presence of entities and hashtags along with event types and titles. Next, we present our proposed approach in detail and our findings.

2 TREC-IS

For standardized evaluations of systems, TREC-IS provided participants with training and test datasets, comprised of three components: the ontology of high-level information types, a collection of crisis-event descriptions, and the tweets for each event to be categorized. The participant TREC-IS systems are intended to produce two outputs for crisis-related social media content:

1. Classifying tweets by *information type*, where each tweet should be assigned as many categories as are appropriate;
2. Ranking tweets by their criticality (*priority*).

TREC-IS provided multiple Twitter datasets collected from a range of past wildfires, earthquakes, floods, typhoons/hurricanes, bombings, and shooting events. The information types as either top-level intent, high-level or low-level.

3 EMERGENCY TWEETS CLASSIFICATION

Our proposed method is a Transformer-based approach that consists of a hierarchical architecture that consists in a hierarchical, multitask learning approach, with a fine-tuned encoder based on RoBERTa,

as shown in Figure 2. This model includes a Transformer (Vaswani et al., 2017) encoder with two Transformer layers on top of the RoBERTa pre-trained model which acts as a task adapter (Pfeiffer et al., 2020) for detecting emergency tweets. The output RoBERTa token representations are fed into a stack of Transformer layers (Vaswani et al., 2017) and then concatenated with the CLS representation, which afterward is fed into two output layers for classification. The attention modules in the Transformer layers adapt not only to the task but also to the noisy input with non-standard or out-of-vocabulary tokens that are specific to social media language (Boros et al., 2020).

In detail, let $\{x_i\}_{i=1}^l$ be a token input sequence consisting of l words, denoted as $\{x_i\}_{i=1}^l = \{x_1, x_2, \dots, x_i, \dots, x_l\}$, where $x_i (1 \leq x_i \leq l)$ refers to the i -th token in the sequence of length l . We first apply a pre-trained language model as *encoder* for further fine-tuning. The output is $\{h_i\}_{i=1}^l, H_{[CLS]} = \text{encoder}(\{x_i\}_{i=0}^l)$ where $\{h_i\}_{i=1}^l = [h_1, h_2, \dots, h_l]$ is the representation for each i -th position in x token sequence and $h_{[CLS]}$ is the final hidden state vector of $[CLS]$ as the representation of the whole sequence x .

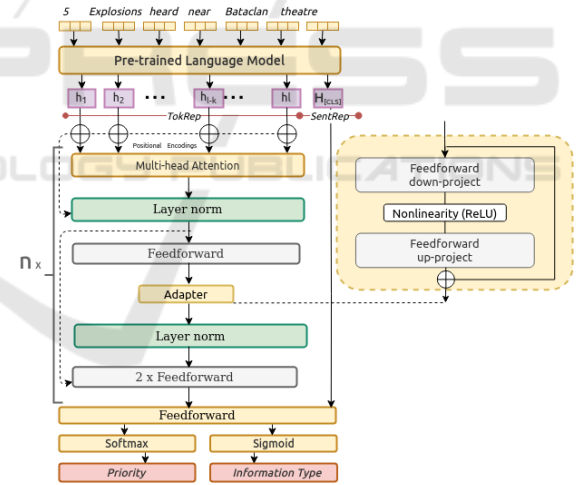


Figure 2: The main architecture of the our proposed model.

From now on, we refer to the *Token Representation* as $\text{TokRep} = \{x_i\}_{i=1}^l$ that is the token input sequence consisting of l words, and the *Sentence Representation* as the $\text{SentRep} = H_{[CLS]}$ that is the representation of the whole sequence.

Next, in order to adapt the model to detecting events in noisy social media posts, we need to add the Transformer encoder with several Transformer layers over a sequential representation, thus on the *TokRep*.

The Transformer encoder contains a number of Transformer layers that takes as input the matrix $H =$

$\{h_i\}_{i=1}^l \in R_{l \times d}$ where d is the input dimension (encoder output dimension).

A Transformer layer includes a multi-head self-attention $Head(h)$: $Q^{(h)}, K^{(h)}, V^{(h)} = HW_q^{(h)}, HW_k^{(h)}, HW_v^{(h)}$ and $MultiHead(H) = [Head^{(1)}, \dots, Head^{(n)}]W_O$ ⁴

where n is the number of heads and the superscript h represents the head index. Q_t is the query vector of the t -th token, j is the token the t -th token attends. K_j is the key vector representation of the j -th token. The $Attn$ softmax is along the last dimension. $MultiHead(H)$ is the concatenation on the last dimension of size $R^{l \times d}$ where d_k is the scaling factor $d_k \times n = d$. W_O is a learnable parameter of size $R^d \times d$. Finally, by combining the position-wise feed-forward sub-layer and multi-head attention, a feed-forward layer is defined as: $FFN(f(H)) = \max(0, f(H)W_1)W_2$ where W_1, W_2 are learnable parameters and \max is the $ReLU$ activation. $W_1 \in R^{d \times d_{ff}}$, $W_2 \in R^{d_{ff} \times d}$ are trained projection matrices, and d_{ff} is a hyperparameter. The task adapter is applied at this level on $TokSep$. The task adapter at each layer consists of a down-projection $D \in R^{h \times d}$ where h is the hidden size of the Transformer model and d is the dimension of the adapter, also followed by a $ReLU$ activation and an up-projection $U \in R^{d \times h}$ at every layer. This task adapter has the only parameters that are updated when training on this downstream task and aims to capture knowledge that is task-specific in regards to non-canonical language in tweets.

Next, we concatenate the obtained sequential transformation and the representation of the sequence: $TokRep + SentRep = [FFN(f(H)), h_{[CLS]}]$.

Finally, the learning of the model is conducted end-to-end by optimizing two objectives corresponding to *information type* classification and *priority* classification respectively: $L_{Info-Type} = -\sum_i^n t_i \log(pt_i)$ and $L_{Priority} = -\sum_i^m c_i \log(pc_i)$, where n and m are the number of classes for each classification task, and t_i and c_i are the true labels, and pt_i and pc_i , the predictions. Finally, we calculate the total loss: $L = \lambda L_{Info-Type} + (1 - \lambda)L_{Priority}$.

Hashtag Augmentations. Twitter trends emerge rapidly or unexpectedly and gain viral traction due to hashtags. A hashtag is a combination of keywords preceded by the # symbol, excluding any spaces or punctuation. We pre-process them by obtaining the separate keywords with a simple rule that tokenizes the hashtag at the encounter of an uppercase letter (e.g., *#FranceShooting* becomes *# France Shooting*).

⁴We leave out the details that can be consulted in (Vaswani et al., 2017)

Entity Augmentations. Entities can be very helpful when aid is needed in specific locations and time frames, etc. This can be done by raising the importance of tweets that are related to a person, a product, an organization, etc. We used a statistical out-of-the-box entity recognition system⁵ that can identify a variety of named and numeric entities including locations (LOC), organizations (ORG), and persons (PER).

How Do We Augment the Input? For exploring the hashtags and the entities, we implemented the pre-trained language model with *EntityMarkers* (Soares et al., 2019; Moreno et al., 2021; Moreno et al., 2020; Boros et al., 2021). First, our model extends the RoBERTa (Liu et al., 2019) model applied to text classification and we add two dense linear layers with softmax activation for the separate tasks: *information type* and *priority*. Then, we augment the input tweet with a series of special tokens (e.g., $\langle \# \rangle$, $\langle \text{entity type} \rangle$). Thus, if we consider a sentence $x = [x_0, x_1, \dots, x_n]$ with n tokens, we augment x with two reserved word pieces to mark the beginning and the end of each event argument mention in the sentence, as in the following example: $\langle \# \rangle \text{BREAKING} \langle / \# \rangle ! 5 \text{ Explosions heard near Bataclan theater} \langle \# \rangle \text{fusillade} \langle / \# \rangle \langle \# \rangle \langle \text{GPE} \rangle \text{Paris} \langle / \text{GPE} \rangle \langle / \# \rangle \langle \# \rangle \langle \text{GPE} \rangle \text{France} \langle / \text{GPE} \rangle \text{Shooting} \langle / \# \rangle$.

Event Metadata. Additionally, we concatenate the augmented tweet text with the event title and type found in the topic description (e.g., *bombing* and *2015 Paris attacks*).

4 EXPERIMENTS & RESULTS

The TREC-IS dataset has a number of emergency events covering different types: earthquakes, tropical storms (e.g., hurricanes), mass violence (e.g., shootings, bombings), public health emergencies (e.g., epidemics), etc. Each incident is accompanied by a brief topic statement that contains the event title (e.g., 2015 Paris attacks), event type (e.g., bombing), and a narrative or description of the event. The provided dataset consisted of a total of 73,499 tweets that covered 75 topics. We did not perform any pre-processing on the data. In our internal experimental setup, we produced a data split for simulating the official TREC-IS split that contains unknown event types in the test set⁷. In

⁵We used the model provided by spaCy v3.0+⁶ (Honibal and Montani, 2017).

⁷The official test set is not available.

Table 1: Detailed results for all our proposed models in comparison with a *Simple* model, which is the RoBERTa model with [*SentRep* and *TokRep*] +/- Transformer layers that uses no enhancement/augmentation.

Approach	nDCG	Info-Type			Priority			
	@100	F1(Act)	F1(All)	Accuracy	F1(Act)	F1(All)	R(Act)	R(All)
Baselines & other models								
BERT <i>SentRep</i> (Wang et al., 2021)	0.4467	0.0458	0.1202	0.7296	0.2711	0.1750	0.2440	0.1699
RoBERTa <i>SentRep</i>	0.4256	0.0439	0.1366	0.3644	0.1100	0.1826	0.0000	0.0000
RoBERTa <i>TokRep</i>	0.4777	0.0462	0.1315	0.6411	0.2247	0.2645	0.1480	0.1494
Our models								
RoBERTa <i>SentRep</i> + <i>TokRep</i>								
Simple	0.5015	0.0942	0.1726	0.8804	0.2711	0.2887	0.2929	0.2186
Ent	0.4967	0.0959	0.1846	0.8824	0.3006	0.3110	0.2598	0.2098
#	0.4873	0.0860	0.1773	0.8817	0.2638	0.2786	0.2504	0.1880
<i>EvtNameType</i> + #	0.4902	0.0579	0.1742	0.8766	0.2456	0.2719	0.2117	0.2089
Ent + #	0.4995	0.0484	0.1638	0.8820	0.2510	0.2751	0.2476	0.1832
Ent+ <i>EvtNameType</i> + #	0.4917	0.0083	0.0993	0.8795	0.2320	0.2755	0.1364	0.1702
RoBERTa <i>SentRep</i> + <i>TokRep</i> + 1×Transformer								
Simple	0.4831	0.0446	0.1085	0.7260	0.3117	0.3326	0.1512	0.2024
Ent	0.4761	0.1625	0.2385	0.8786	0.3115	0.3211	0.3647	0.2423
#	0.4825	0.1813	0.2291	0.8773	0.3100	0.3249	0.2349	0.1964
<i>EvtNameType</i> + #	0.5414	0.1772	0.2534	0.8829	0.2859	0.3066	0.3427	0.2150
Ent + #	0.4803	0.0704	0.1428	0.8770	0.2799	0.2923	0.2931	0.1904
Ent+ <i>EvtNameType</i> + #	0.5052	0.1548	0.2308	0.8824	0.2982	0.3170	0.2764	0.2175
RoBERTa <i>SentRep</i> + <i>TokRep</i> + 2×Transformer								
Simple	0.4835	0.1419	0.2139	0.8755	0.3059	0.3228	0.3006	0.2368
Ent	0.4862	0.2058	0.2483	0.8761	0.2977	0.3122	0.3889	0.2413
#	0.4737	0.1555	0.2221	0.8773	0.3038	0.3229	0.2236	0.2042
<i>EvtNameType</i> + #	0.5060	0.1793	0.2552	0.8840	0.3171	0.3296	0.3490	0.2772
Ent + #	0.4750	0.0675	0.1872	0.8818	0.2450	0.2740	0.1327	0.1522
Ent+ <i>EvtNameType</i> + #	0.4946	0.1577	0.2449	0.8830	0.3456	0.3290	0.2513	0.1924
RoBERTa <i>SentRep</i> + <i>TokRep</i> + 3×Transformer								
Simple	0.4690	0.2102	0.2428	0.8768	0.2670	0.3064	0.1423	0.1779
Ent	0.4779	0.1415	0.2419	0.8768	0.2746	0.3090	0.3073	0.2205
#	0.4892	0.1899	0.2345	0.8777	0.3252	0.3317	0.2670	0.2147
<i>EvtNameType</i> + #	0.5145	0.1759	0.2437	0.8842	0.2567	0.2997	0.3556	0.2416
Ent + #	0.4635	0.1143	0.1647	0.8809	0.2492	0.2801	0.3291	0.2583
Ent + <i>EvtNameType</i> + #	0.4973	0.1743	0.2529	0.8813	0.2954	0.3209	0.2594	0.2345

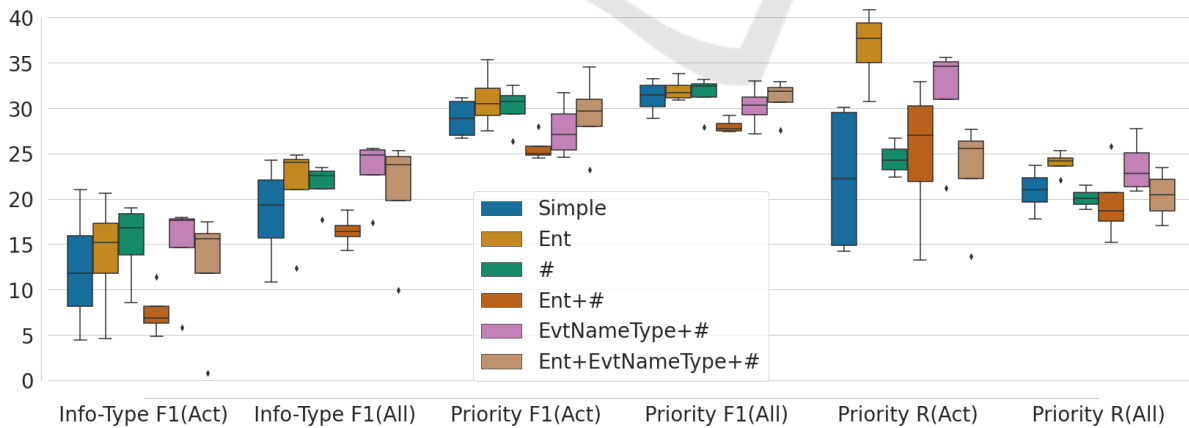


Figure 3: The distribution of the performance scores for six different types of input augmentations.

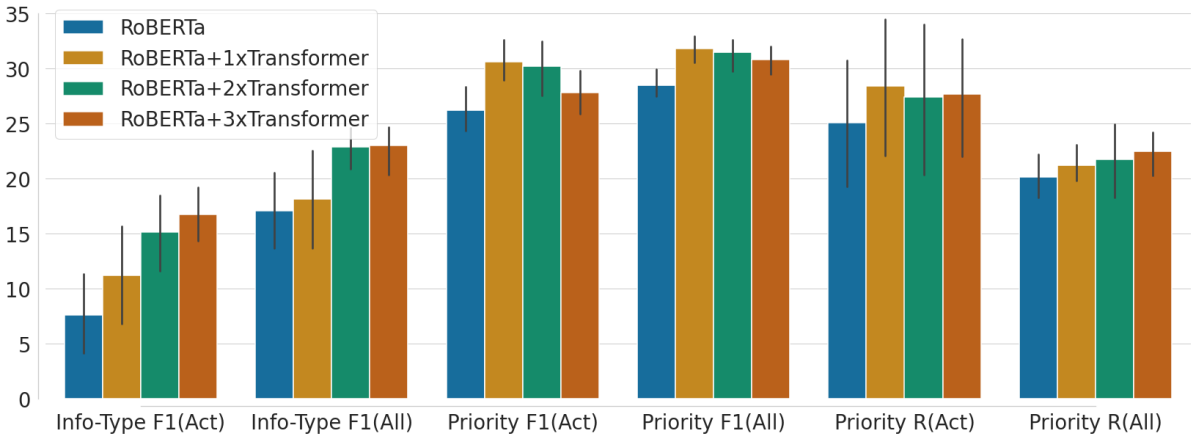


Figure 4: The distribution of the performance scores for RoBERTa+n×Transformer.

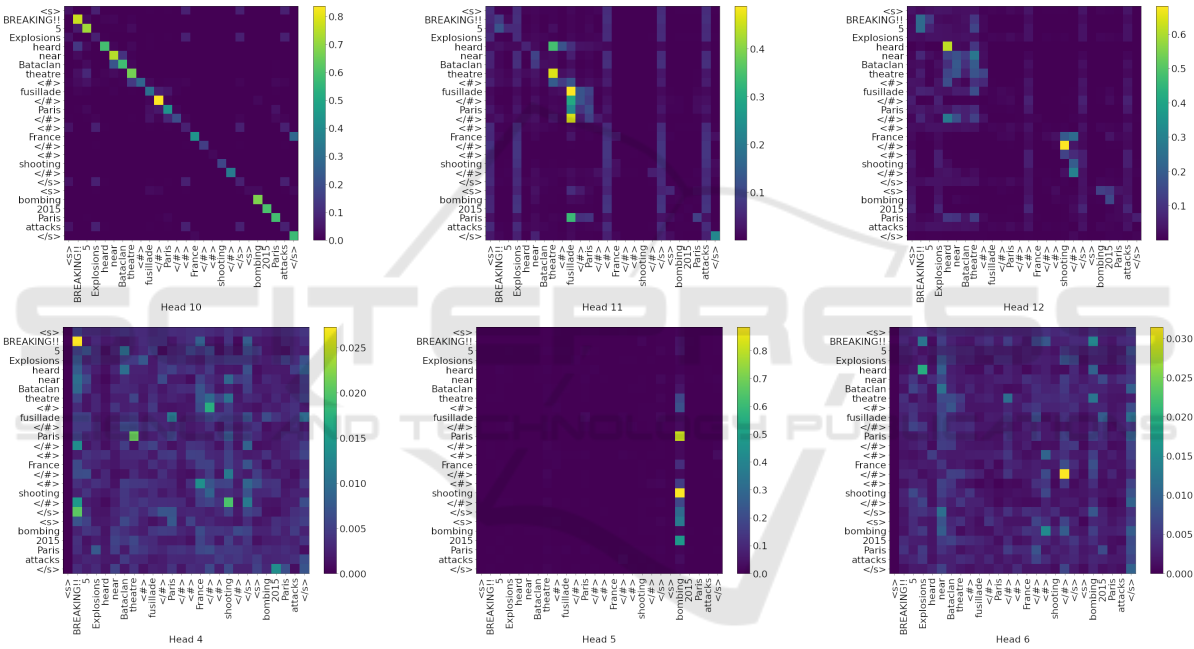


Figure 5: The explorations of the attention behavior for RoBERTa+1×Transformer hashtags and event metadata.

order to ensure this, for the training set, we selected the event types from 0 to 64 (60,577 tweets), and from 65 to 75 (12,923 tweets), for the test set⁸. Thus, in this manner, no events are overlapped in the train and test set.

4.1 Evaluation

To evaluate the performance of such systems, the following two groups of metrics were proposed by TREC: for *information type*, its overall F1(All) score, macro-averaged across all types and micro-averaged

across events, and its F1(Act) score among the *actionable* types. For *prioritization*, its overall prioritization error is considered, micro-averaged across events, F1(Act), and macro-averaged across all types, F1 (All), and priority scores correlational performance, R(All) and R(Act). We experimented with four models based on RoBERTa + n×Transformer with $n \in \{0, 1, 2, 3\}$ (when $n = 0$, the token representations are not used), and six different types of input augmentations: Simple (no augmentations), Ent (text with marked entities), # (text with marked and pre-processed hashtags), Ent+# (the previous two together), EvtNameType+# (hashtags and the event type and title), and Ent+EvtNameType+#.

⁸More details about the data can be found at http://dcs.gla.ac.uk/~richardm/TREC_IS/2020/data.html.

4.2 Input Augmentations

Figure 3 shows the distribution of the performance scores for the six different types. For *information type* classification (Info-Type), we observe that, while the highest F1(Act) and F1(All) are obtained when hashtags and entities are marked separately, the lowest scores are obtained when they are combined. We also notice that augmenting the tweets with the event title and type, besides hashtags, outperforms the models that use entities or no augmentations. However, when the augmentations are performed altogether, the distribution of the scores is negatively skewed, with the majority of the models underperforming. The results without any augmentation tend to vary the most, although the median value remains stable. When detecting the *priority*, we observe the same tendencies regarding the entities, while the events and hashtags generally perform the best. When including only entities, the R(Act) scores outperform considerably the other types, while when adding the pre-processed hashtags and the event titles and types, the R(All) surpasses the others.

4.3 Transformer Adapters

Figure 4 shows the distribution of the performance scores for RoBERTa+ n ×Transformer. For the prediction of the *information type*, we notice that F1(All) and F1(Act) are generally lower when no additional Transformer layer is used, and increase proportionally with the number of Transformer layers. For *priority*, however, the highest results were brought by adding just one Transformer, while generally overfitting with more than one.

For further understanding of the impact of the Transformer adapters and the input enhancements, we visualize the last three attention matrices for two selected layers, the eleventh layer of RoBERTa and the first Transformer adapter. Based on the Figure 5 above we observe that there is a high attention set along the diagonals and on an informative tokens such as *BREAKING!!* and the event metadata (*bombing* and *2015 Paris attacks*). For the Transformer layer, the attention finds correlations between *Paris* and *theater*, between hashtags markers <#> and other informative tokens such as *France* or *shooting*. Such a pattern could indicate that the additional Transformer layers are able to identify correlations between factual impact factors to detecting emergency events on social media.

4.4 SentRep versus TokRep

Finally, in Table 1, we compare our methods with a recent work (Wang et al., 2021) that consists in a multitask BERT with a *SentRep*, with a regression task *priority* and a classification task for *information type*. We also compare two base models (with *SentRep*), for analyzing the importance of adding *TokRep*. We can observe that, generally, the classification of *information type* mostly benefits the *TokRep* and *SentRep* together with more than two additional Transformer layers applied to them, obtaining F1 scores higher than 0.20. While we notice that *SentRep*, for a regression task (Wang et al., 2021), can be enough for predicting the *priority*, we agree more towards the fact that the most important and impactful factor is the *TokRep*, that gain a considerable increase when compared only with *SentRep*.

Finally, our best results are revealed in Table 1 in comparison with the state-of-the-art models proposed by (Wang et al., 2021) and our baseline without any enhancements.

5 CONCLUSIONS AND FUTURE WORK

Our experiments showed that considering the token sequence encoded by additional adapted Transformer layers augmented with either entities or event metadata could bring promising improvements in detecting the criticality of events in social media posts. Further work will be focused on performing different ablation studies (e.g., number and size of the attention heads) and error analysis. This work could open new ventures toward more effective emergency and actionable event detection.

REFERENCES

- Boros, E., Hamdi, A., Pontes, E. L., Cabrera-Diego, L.-A., Moreno, J. G., Sidere, N., and Doucet, A. (2020). Alleviating digitization errors in named entity recognition for historical documents. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 431–441.
- Boros, E., Moreno, J. G., and Doucet, A. (2021). Event detection with entity markers. In Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., and Sebastiani, F., editors, *Advances in Information Retrieval*, pages 233–240, Cham. Springer International Publishing.
- Buntain, C., McCreadie, R., and Soboroff, I. (2020). Incident streams 2020: Trecis in the time of covid-19.

- 18th International Conference on Information Systems for Crisis Response and Management.*
- Choi, W.-G., Jo, S.-H., and Lee, K.-S. (2018). Cbnu at trec 2018 incident streams track. In *TREC*.
- Chy, A. N., Siddiqua, U. A., and Aono, M. (2018). Neural networks and support vector machine based approach for classifying tweets by information types at trec 2018 incident streams task. In *TREC*.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Dusart, A., Hubert, G., and Pinel-Sauvagnat, K. (2019). Irit at trec 2019: Incident streams and complex answer retrieval tracks. In *Text REtrieval Conference*. National Institute of standards and Technology (NIST).
- García-Cumbreras, M. Á., Díaz-Galiano, M. C., García-Vega, M., and Jiménez-Zafra, S. M. (2018). Sinai at trec 2018: Experiments in incident streams. *Weather*, 38(3):14.
- Honnibal, M. and Montani, I. (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *Unpublished software application*.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pre-training approach. *arXiv preprint arXiv:1907.11692*.
- McCreadie, R., Buntain, C., and Soboroff, I. (2019). Trec incident streams: Finding actionable information on social media. *Proceedings of the 16th ISCRAM Conference*.
- McCreadie, R., Buntain, C., and Soboroff, I. (2020). Incident streams 2019: Actionable insights and how to find them. *Proceedings of the International ISCRAM Conference*.
- Mishra, A. and Pal, S. (2019). Iit bhu at trec 2019 incident streams track. In *TREC*.
- Miyazaki, T., Makino, K., Takei, Y., Okamoto, H., and Goto, J. (2018). Nhk strl at trec 2018 incident streams track. In *TREC*.
- Miyazaki, T., Makino, K., Takei, Y., Okamoto, H., and Goto, J. (2019). Label embedding using hierarchical structure of labels for twitter classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6317–6322.
- Moreno, J. G., Boros, E., and Doucet, A. (2020). Tlr at the ntcir-15 finnum-2 task: Improving text classifiers for numeral attachment in financial social data. In *Proceedings of the 15th NTCIR Conference on Evaluation of Information Access Technologies, Tokyo Japan*, pages 8–11.
- Moreno, J. G., Doucet, A., and Grau, B. (2021). Relation classification via relation validation. In *Proceedings of the 6th Workshop on Semantic Deep Learning (SemDeep-6)*, pages 20–27.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Pfeiffer, J., Vulić, I., Gurevych, I., and Ruder, S. (2020). MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.
- Soares, L. B., FitzGerald, N., Ling, J., and Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Wang, C. and Lillis, D. (2021). Ucd-cs at trec 2021 incident streams track. *arXiv preprint arXiv:2112.03737*.
- Wang, C., Nulty, P., and Lillis, D. (2021). Transformer-based multi-task learning for disaster tweet categorisation. *arXiv preprint arXiv:2110.08010*.