University of Glasgow Terrier Team (uogTr) at the TREC 2020 Incident Streams Track

Alexander J. Hepburn
University of Glasgow, UK
a.hepburn.1@research.gla.ac.uk

Richard McCreadie
University of Glasgow, UK
richard.mccreadie@glasgow.ac.uk

ABSTRACT

In this paper, we describe runs submitted on behalf of the University of Glasgow Terrier Team (uogTr) for the TREC 2020 Incident Streams track. We detail our approach to addressing the challenges present in the classification of crisis and disaster management data in unstructured text. In particular, we explore the usage of pretrained ELMo embeddings alongside descriptive metadata-level and event-level features for classification. We also utilise algorithms incorporating undersampling techniques in order to mitigate the significant class imbalance in the dataset. We submitted a total of three official runs to the 2020A track: `ELMO_ALL_BRF`, `ELMO_ALL_EEC`, and `ELMO_ALL_TFIDF_BRF` with varying features and classifiers used. Our results show that our run, `ELMO_ALL_BRF` shows competitive performance, performing above the median across a number of track-specific metrics.

1 INTRODUCTION

The Terrier team (uogTr) at the University of Glasgow participated in the TREC 2020 Incident Streams track. We experimented with a variety of approaches and features that resulted in three official submitted runs to the 2020A edition of the track, and an unsubmitted run for the 2020B edition of the track. In particular, we explore the complexities of classification for crisis and disaster datasets as part of ongoing work into event-centric information extraction. We approached the problem by leveraging advances made in deep, contextual word representations and utilised ELMo[5] embeddings to generate word embeddings for tweets. We use these per-word embeddings via average pooling to produce a single embedding for each tweet. Following an analysis of the characteristics of actionable tweets in the dataset, we also incorporated the use of a pretrained entity recognition pipeline to identify location information within the tweet. Furthermore, we leveraged associated tweet and event metadata to produce a number of numerical and event-level features, the latter of which proved to be especially valuable in disambiguating actionable information from general discussion. For learning methods, we experimented with both Balanced Random Forest[1] and EasyEnsemble[3] models, which both use random undersampling during training, to address the significant class imbalance present in the dataset.

The remainder of this paper is structured as follows: Section 2 will discuss the features created from the tweet text and the preprocessing steps therein; Section 3 will discuss the numeric features present in our work, generated from tweet metadata and the transformation steps we carried out to achieve this; Section 5 outlined our submitted runs for the 2020A edition of the track; Section 6 details our results compared with the median and best-performing results; Section 7 includes a brief discussion of those results; and Section 8 includes our concluding remarks.

Source	Information	#Tweets	%
Tweet Text	Terms/Phrases	164	97%
	Location	150	88%
	Event Mention	34	20%
	Time Mention	10	6%
	Person Mention	10	6%
Linked Content	Article/Web Page	83	49%
	Tweet	21	12%
	Image	8	5%
	Video	5	3%
Author	Name/Username	7	4%
Regional Context Needed		42	25%
Tweet is Out of Date		41	24%

Table 1: Statistics for what makes a tweet critical from analysis of the 2019-B critical tweets. Reconstructed from [4].

2 TEXT-BASED FEATURES

Our approach is inspired by an analysis of the Critical tweets present in the 2019B dataset by McCreadie et al. [4]. As shown in the Table 1 below that summarizes the outcome of that work, we observe that the majority (97%) of tweets contained terms and/or phrases that indicate actionable information was embedded in the text content. Users on Twitter are constrained to a maximum of 280 characters per tweet and as such, tweets often contain contractions, abbreviations, and colloquial language in order for users to express themselves within these restrictions. Hence, we normalise our text and utilise external dictionaries to reduce our embedding space as much as possible, specifically, a slang dictionary and a contraction dictionary. We use pretrained, high-dimensional ELMo word embeddings to more accurately capture the complex syntax and semantics often found in short text representations. These embeddings then go through an average pooling process to produce a single, document vector for each tweet. Prior to this, mentions, URLs, and text containing over two identical consecutive characters were removed during preprocessing. Hashtags were left as they appeared in the original text, as they are often used in conjunction with other hashtags or relevant tokens to identify latent topics.
In order to further enrich our training feature vector, we must examine what is present within the text that distinguishes these tweets from those that are less critical to emergency response. Location is explicitly mentioned in 88% of Critical priority tweets, so we incorporated a named entity recognition module to create a binary representation of the presence of a location mentioned within the tweet text; to achieve this, we use a pre-trained named entity recognition pipeline provided by the SparkNLP[1] package. We

[1] https://nlp.johnsnowlabs.com/

found that the addition of this binary feature provided a significant boost in the performance of each run.

3 NUMERICAL FEATURES

We generate a number of numeric features from additional metadata provided by the JSON structure of each tweet as follows:

- The number of hashtags present
- The presence of URLs (binary)
- The presence of additional media elements such as pictures and videos (binary)

We also incorporate an event-level feature found in the per-topic metadata files supplied by the track. The training data provided by the Incident Streams track is split by events, and each event corresponds to a *type* within a provided in the *topics* file, which provides extended metadata on each event. The types present across both training and test sets are: EARTHQUAKE, FLOOD, TYPHOON, WILDFIRE, SHOOTING, BOMBING. We leveraged this information to include a one-hot encoded feature representing each event subtype.

4 EXPERIMENTAL SETUP

Our models were trained on the 2019A, and 2019B datasets from previous editions of the track. This collection consists of 15,673 tweets after cleaning and filtering stages. We approach the Information Type classification task as a multi-label classification problem. As such, we transform our multi-label classification problem with n labels into n separate, binary classification problems using each of the aforementioned models as our base classifier. Throughout experimentation, we maintain the following hyperparameters: In this case of Balanced Random Forest, we set the number of estimators (`n_estimators`) to 100, no limit on the max depth of the tree (`max_depth`); in the case of our EasyEnsemble classifier, we set our number of estimators `n_estimators`) to 10 and set `warm_start` to `False`, which results in fitting an entirely new ensemble at each step of the algorithm. The implementations of these models are provided by the imbalanced-learn[2] Python package.

5 SUBMITTED AND UNSUBMITTED RUNS

Considering the aforementioned features, we structure our experimentation in order to address the following research questions:

(1) Is there significant difference in performance between employing the use of Balanced Random Forest and EasyEnsemble classifiers?
(2) Does the inclusion of average pooled TF-IDF vectors to represent each document benefit overall classification performance?

5.1 Submitted Runs

The maximum number of submissions a participant can make to the track is four and as such, a total of three separate runs were submitted to the first two tasks of 2020A edition of the track:

(1) **`ELMO_ALL_BRF`**: Our first submitted run uses the Balanced Random Forest algorithm developed by Chen et al.[1]. It is an adaption of the Random Forest algorithm and attempts to balance the tree at each step of the algorithm by undersampling from the majority class, we utilise this to mitigate the influence and effect of lower criticality, irrelevant tweets that are disproportionately common in our data.
(2) **`ELMO_ALL_EEC`**: Our second run is very similar to our first in that it includes the same set of features. However, it uses the boosting algorithm EasyEnsemble developed by Liu et al.[3] which uses a combination of AdaBoost[2] learners and undersampling to generate synthetic samples characteristic of those belonging to our minority classes and to randomly undersample those from majority classes.
(3) **`ELMO_ALL_TFIDF_BRF`**: Noticing a performance increase with Balanced Random Forest, we decided to submit a third run using the same algorithm. We decided to include an additional average pooled TF-IDF vector in the hopes of mitigating the drop in accuracy of our models.

6 RESULTS

Table 2 reports the performance of our submitted runs in comparison to the TREC Best and Median systems. We abbreviate each track metric as follows: "HAAW" means High Priority Accumulated Alert Worth, "AAAW" means Accumulated Alert Worth (All), "ITAct" means Information Type Positive F1-score (Actionable), "ITAll" means Information Type Positive F1-score (All), "ITAcc" means Information Type Accuracy, "PAct" means Priority F1-score (Actionable), and "PAll" means Priority F1-score (All).

An analysis of the results table (Table 2) shows that the introduction of a TF-IDF feature in `elmo_all_tfidf_brf` seemed to be more effective in alerting the user of relevant posts. It also outperformed other runs in overall Information Type Accuracy, suggesting that this approach creates a better, general representation of the underlying data. However, this run fell short in terms of classifying the criticality of tweets, both actionable and overall. Our Balanced Random Forest approach seems to generally outperform our EasyEnsemble classifier across the board, however EasyEnsemble seems to be slightly more performant in overall alerting. We notice both algorithms struggle from a particularly low type categorisation accuracy, and upon further inspection, it seems our system is overly generous when assigning relevant labels to any single tweet. As expected, overall accuracy from each submission increases with the reduced set of information type labels. However, we notice that this brings a significant decrease in both information type and priority F1-score across the board. Despite the increase in performance across ranking and alerting metrics, the inclusion of a TF-IDF-based document vector negatively impacts our type categorisation and priority estimation scores in the first task. Interestingly, this is not reflected in the second task, suggesting that the benefit of this additional feature is lost as the number of labels increases.

7 DISCUSSION

Comparison with other methods. Our proposed method is amongst the top scoring methods submitted to the 2020A edition of the track. Our TF-IDF run with Balanced Random Forest is the second highest scoring method for the first task with regard to alerting metrics (for both high and low priority tweets), as is our baseline Balanced

[2] https://imbalanced-learn.org

Table 2: TREC-IS Performance on the 2020-A events

Task 1: 25 Information Types										
Run			Ranking	Alerting		Type Categorization			Priority	
ID	Learner	Text Encoding	NDCG	HAAW	AAAW	ITAct	ITAll	ITAcc	PAct	PAll
TREC Best	-	-	0.4866	0.2121	-0.0847	0.1674	0.2089	0.9403	0.2642	0.2800
TREC Median	-	-	0.4235	-0.4488	-0.2451	0.0792	0.1380	0.8978	0.1524	0.2076
elmo_all_2020_brf	BRF	ELMO	0.4212	-0.2628	-0.2028	0.0933	0.1438	0.7502	0.2630	0.2575
elmo_all_2020_eec	EE	ELMO	0.4235	-0.2674	-0.1863	0.0769	0.1371	0.7284	0.2076	0.2351
elmo_all_tfidf	BRF	TF-IDF+ELMO	0.4301	-0.2090	-0.1705	0.0884	0.1380	0.7529	0.1651	0.1929

Task 2: 12 Information Types										
Run			Ranking	Alerting		Type Categorization			Priority	
	Learner	Text Encoding	NDCG	HAAW	AAAW	ITAct	ITAll	ITAcc	PAct	PAll
TREC Best	-	-	0.4864	0.2121	-0.0847	0.1695	0.2079	0.9535	0.2077	0.2612
TREC Median	-	-	0.3919	-0.5101	-0.2598	0.0563	0.0886	0.8773	0.1187	0.1417
elmo_all_2020_brf	BRF	ELMO	0.4222	-0.2628	-0.2028	0.0933	0.1113	0.8035	0.1821	0.1672
elmo_all_2020_eec	EE	ELMO	0.4235	-0.2674	-0.1863	0.0770	0.993	0.7835	0.1370	0.1612
elmo_all_tfidf	BRF	TF-IDF+ELMO	0.4297	-0.2090	-0.1705	0.0881	0.1036	0.8072	0.1737	0.1913

Random Forest run for the Priority F1-score over actionable classes. As mentioned previously, our accuracy is among the lowest in the table. This is something we hope to rectify in future editions of the track by experimenting with different feature generation methods and algorithm decisions.

Reflections on proposed method. Given the significant boost in performance with the addition of event types as a feature during training, we would like to experiment with the inclusion of more general event-level information for each tweet. We would also like to investigate the use of entity linkage with external sources to enrich the knowledge of the model during training.

8 CONCLUSIONS

In this paper, we proposed a method of classifying crisis and disaster tweets to assist emergency response. We confirmed its effectiveness via evaluation in the 2020A edition of the TREC Incident Streams track. Building on and further developing the quality of our runs by leveraging external datasets on crises, and experimenting with various complementary tasks is something we have left for future work.

ACKNOWLEDGEMENTS

We would like to thank the organisers of the TREC Incident Streams track for providing datasets, evaluating our work, and facilitating a community around the role of information retrieval in crisis and emergency management.

REFERENCES

[1] Chao Chen and Leo Breiman. 2004. Using Random Forest to Learn Imbalanced Data. *University of California, Berkeley* (01 2004).
[2] Yoav Freund and Robert E. Schapire. 1999. A Short Introduction to Boosting.
[3] X. Liu, J. Wu, and Z. Zhou. 2009. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 2 (2009), 539–550. https://doi.org/10.1109/TSMCB.2008.2007853
[4] Richard McCreadie, Cody Buntain, and Ian Soboroff. 2020. Incident Streams 2019: Actionable Insights and How to Find Them. In *17th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2020)*. http://eprints.gla.ac.uk/210955/
[5] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR* abs/1802.05365 (2018). arXiv:1802.05365 http://arxiv.org/abs/1802.05365