University of Glasgow Terrier Team (uogTr) at the TREC 2021 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 detail our approach as part of the runs submitted on behalf of the University of Glasgow Terrier Team (uogTr) for the 2021-A/B edition of the Incident Streams track. Our approach employs the use of transfer learning between component labels of the dataset; more specifically, we decompose the traditional multi-label approach and investigate the relationship between each label as a binary classification task. We submit a total of three official runs to the 2021-A/B edition of the track, namely: `uogTr-01-pw`, `uogTr-02-pwcoocc`, and `uogTr-04-coocc`. Our results show that there exists potential for performance increase through transfer learning.

1 INTRODUCTION

On behalf of the Terrier Team (uogTr) at the University of Glasgow, we have submitted a total of three official runs to the TREC 2021 Incident Streams track. We experimented with a variety of different approaches as part of ongoing work into task relatedness in transfer learning. We approached the problem by decomposing the multi-label approach into that of separate binary classification tasks, and investigated the degree of synergy between these tasks through inductive transfer. As our base model, we utilise the pretrained, case-sensitive $\text{BERT}_{\text{BASE}}$ model provided by HuggingFace[1] library.

As mentioned previously, our submissions to this track were as a result of ongoing work into the investigation of the quality of task relatedness, using prior editions of the Incident Streams dataset. We performed an exhaustive search over task and hyper-parameter combinations in order to determine the best-performing combinations for transfer. However, as the number of tasks and parameters increase, the number of possible combinations quickly becomes computationally infeasible to train. In order for this to be achievable for our work, we limited our scope to the **Task 2** formulation from previous editions of TREC-IS, which restricted the number of tasks to 12, listed in Fig. 1.

Beginning with the BERT model, we then fine-tuned each pair of tasks in succession (i.e. BERT→Source Task→Target Task). To estimate the priority of each document, we used a simple logistic regression layer in place of our classification head.

The remainder of this paper is structured as follows: Section 2 will discuss the process involved in the transformation of the multi-label ontology to separate binary tasks, the details of the dataset included, the parameters used, and our preprocessing steps; Section 3 discusses our approach as part of our work into transfer learning and task relatedness; Section 4 outlined our submitted runs for the 2021-A/B edition of the track and the differences between them in approach; Section 5 details our results compared with other participants in the track; Section 6 includes a brief, reflective discussion of our results; and finally, Section 7 includes our concluding remarks.

[1] https://huggingface.co/bert-base-cased

Figure 1: TREC-IS Information Type Ontology, Task 2 Formulation.

Intent Type	Information Type
Request	Goods Services
	Search And Rescue
	Information Wanted
Call To Action	Volunteer
	Move People
Report	First Party Observation
	Emerging Threats
	New Sub-Event
	Multimedia Share
	Service Available
	Location
Other	Any

2 EXPERIMENTAL SETUP

Dataset. In order to compute the best pairs from the prior **Task 2** formulations of the track, we initially investigated this using the 2019-A/B (17.2k documents) and 2020-A (6.6k documents) editions of the track for training and testing, respectively.

For submission to the 2021 edition of the track, we add the 2020-A and 2020-B labelled datasets to our training set, giving us a total of 70.4k documents for training. We further split our training set into training (90%) and validation (10%) sets (63.4k v. 7k documents).

Finally, we test on the unlabelled 2021-A collection provided by the track, totalling 1.5M documents.

Preprocessing. We use only the tweet text from each document, capped at 280 characters as features. As we make use of the $\text{BERT}_{\text{BASE}}$ model, we follow much the same preprocessing steps as the original authors, wrapping each input sequence in special tokens `[CLS]` and `[SEP]` which denote the special classification and separator (end-of-input signifier) tokens, respectively. Afterwards, we perform a few additional normalisation steps; we remove URLs, Mentions (@ symbol followed by username of mentioned user), and the "RT:" prefix signifying a retweet.

Model. Using the aforementioned $\text{BERT}_{\text{BASE}}$ model as a starting point, we add a linear layer on top of the network for binary classification. We also use a learning rate scheduler, an AdamW optimiser [2], and add a dropout with a rate of 30%. For the purposes of estimating the priority of each document, our classification layer is replaced by a regression head.

		Inductive Transfer (Source)				Target Parameters			Evaluation Scores	
Target	Model	Transfer-From	LR	#E	B#	LR	#E	B#	Positive F1	Accuracy
New Sub Event	BERT→Target	None	-	-	-	2e-05	4	16	0.0258	0.9604
	BERT→Source→Target	Volunteer (Best)	1e-05	2	32	2e-05	2	32	0.0618 ▲	0.9405
First Party Observation	BERT→Target	None	-	-	-	2e-05	4	32	0.0259	0.9646
	BERT→Source→Target	Move People (Best)	1e-05	2	32	1e-05	1	32	0.1142 ▲	0.9538
Service Available	BERT→Target	None	-	-	-	3e-05	3	16	0.0944	0.9821
	BERT→Source→Target	Other (Best)	1e-05	1	32	1e-05	1	32	0.1095	0.9783
Move People	BERT→Target	None	-	-	-	2e-05	3	32	0.1964	0.9835
	BERT→Source→Target	Information Wanted (Best)	1e-05	2	32	1e-05	2	32	0.2431	0.9850
Emerging Threats	BERT→Target	None	-	-	-	3e-05	2	32	0.2329	0.8323
	BERT→Source→Target	Location (Best)	1e-05	2	32	1e-05	1	32	0.2612 ▲	0.8135
Multimedia Share	BERT→Target	None	-	-	-	2e-05	3	32	0.4356	0.6760
	BERT→Source→Target	Information Wanted (Best)	2e-05	1	32	2e-05	1	32	0.4757 ▲	0.6431
Location	BERT→Target	None	-	-	-	3e-05	2	16	0.5904	0.6939
	BERT→Source→Target	Multimedia Share (Best)	1e-05	1	32	1e-05	1	32	0.6178 ▲	0.7196
Other	BERT→Target	None	-	-	-	5e-05	4	16	0.6831	0.5638
	BERT→Source→Target	Multimedia Share (Best)	2e-05	1	32	1e-05	2	32	0.6853 ▲	0.7187
AVERAGE	BERT→Target	None	-			*Varies*			0.2856	0.8321
	BERT→Source→Target	*Varies*	*Varies*			*Varies*			0.3211	0.8441

Table 1: Information type categorisation performance with and without inductive transfer from a source task. Metrics are micro-averaged across events and range from 0 to 1, higher is better. Statistical significance is measured with McNemar's test. ▲ and ▼ denote significant performance increases and decreases at $p \leq 0.05$

Loss function weighting. After observing poor performance on tasks which had a significantly lower number training examples, we weighted our cross-entropy loss function, optimising for the minority class as follows:

$$w_0 = \frac{|y_M|}{|y_0|}, w_1 = \frac{|y_M|}{|y_1|} \quad \text{where } |y_M| = min(|y_0|, |y_1|) \tag{1}$$

Hyperparameters. We began with the pool of parameters described as *optimal* for BERT in the paper by Devlin et al. [1]. We found, generally, the most stable configuration for all tasks was to set the learning rate to 2e-5, the number of iterations to 2, and the batch size to 32.

However, when experimenting with a transfer learning approach, we discovered that our standard parameters yielded poor results when training in succession, which we hypothesised may be as a result of phenomena such as catastrophic forgetting. As such, we further restricted our pool of parameters to the following[2]:

(1) Learning rate (Adam): 1e-5, 2e-5
(2) Number of epochs: 1, 2

3 APPROACH

Our experimentation on the 12 tasks from the **Task 2** formulation of the 2020-A edition consisted of two separate stages. Firstly, we experimented with hyperparameter tuning and loss function weighting for each task in isolation and secondly, we trained each possible combination of tasks and parameters by fine-tuning in succession (i.e. BERT→Source Task→Target Task).

[2] After observing negligible effects on changing the batch size, we maintained a batch size of 32 for all models.

After discovering that utilising the *optimal* parameters identified in the first stage of our experiments resulted in poor performance, we further reduced our pool of parameters in order to mitigate the effects of excessive training, where potentially useful information may be overwritten.

Our transfer learning experiments yielded some interesting results, shown in Table 1. The categories `InformationWanted`, `SearchAndRescue`, `GoodsServices`, and `Volunteer` showed no observable change in performance through inductive transfer regardless of source information type used. For Emerging Threats and Multimedia Share, we observe an increase in performance of around 8% and 6% respectively, which is a modest increase as these types were already some of the better performing. However, we also observe a staggering performance increase of 191% for First Party Observation. This shows that, for some information types, there is clear scope for improving performance via transfer learning techniques. Indeed, we see an improvement in *Information Type Actionable F1* in 75% of the tasks in the table. Overall, we observed a 12.4% increase across 8 tasks using transfer learning.

We then decided to evaluate our approach for the 2021 edition of the track, adding the 2020-A/B sets to our training set. Since we did not have the available information of best-performing task pairs for the remaining 13 tasks, we determined the suitability of task pairs under the assumption that, for each target task, its top-ranked co-occurring label (in our training set) must exhibit some degree of semantic similarity and may, by extension, be synergistic in training.

TREC-IS 2021-A Results								
Run	Ranking	Type Categorization			Priority			
ID	NDCG@100	ITAct	ITAll	ITAcc	PF1Act	PF1All	PRAct	PRAll
TREC Best	0.6115	0.2815	0.3211	0.8902	0.3060	0.3211	0.4349	0.3585
TREC Median	0.5695	0.2060	0.2823	0.8827	0.2113	0.2175	0.1728	0.2099
uogTr-01-pw	0.3965	0.0983	0.2062	0.8813	0.0301	0.0810	0.0879	0.0654
uogTr-02-pwcoocc	0.3967	0.1657	0.2924	0.8827	0.0301	0.0810	0.0879	0.0654
uogTr-04-coocc	0.3953	0.1657	0.2889	0.8842	0.0301	0.0810	0.0879	0.0654

TREC-IS 2021-B Results								
Run	Ranking	Type Categorization			Priority			
ID	NDCG@100	ITAct	ITAll	ITAcc	PF1Act	PF1All	PRAct	PRAll
TREC Best	0.4791	0.2510	0.2623	0.9067	0.2798	0.2756	0.2302	0.2952
TREC Median	0.4272	0.1842	0.2330	0.8947	0.2107	0.2031	0.1495	0.1993
uogTr-01-pw	0.2928	0.0731	0.1532	0.8916	0.0727	0.1223	0.1375	0.1096
uogTr-02-pwcoocc	0.2945	0.1340	0.2280	0.8982	0.0727	0.1223	0.1375	0.1096
uogTr-04-coocc	0.2934	0.1340	0.2284	0.8977	0.0727	0.1223	0.1375	0.1096

Table 2: TREC-IS Performance on the 2021-A/B events

4 SUBMITTED RUNS

We submitted three separate runs, evaluating all of our individual classifiers as a cohesive system. For each of these submitted runs, we estimated the priority of each document using a logistic regression head in place of our classification layer.

(1) **uogTr-01-pw**: For our first run, we use the parameters which exhibited the highest performance when the 12 tasks were tuned in isolation. For the remaining 13 tasks, we set the learning rate to 2e-5, used 2 training iterations, and a batch size of 32, which generally exhibited the most stable performance across all tasks.
(2) **uogTr-02-pwcoocc**: For our second run, we use the best-performing, combined models from Table 1 for each of the eight target tasks (omitting the four which showed no performance increase from transfer). For the remaining 17 tasks, we used the single-task models with the "standard" hyperparameters of 2e-5, 2, 32 for the learning rate, number of epochs, and batch size, respectively.
(3) **uogTr-04-coocc**: For our third run, we use transfer learning on all tasks, using the models from Table 1 and using the top-ranked co-occurrences for the remaining 13. Taking into account the potential impact on performance from excessive training, we reduce our parameters to 1e-5, 2, 32 for the learning rate, number of training epochs, and batch size, respectively.

5 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: "ITAct" means Information Type Positive F1-score (Actionable), "ITAll" means Information Type Positive F1-score (All), "ITAcc" means Information Type Accuracy, "PF1Act" means Priority F1-score (Actionable), "PF1Act" means Priority F1-score (All), "PRAct" means Priority R (Actionable), and "PRAll" means Priority R (All).

When comparing the *Information Type F1 Actionable* scores between uogTr-01-pw and both uogTr-02-pwcoocc and uogTr-04-coocc, that is to say, the comparison between our system which contained no element of transfer learning and systems that did, we can see a jump of 68% and 83% in performance when using the transfer learning models for the actionable tasks. When we employed transfer learning across all tasks, we see no observable, significant change in the *Information Type F1 All* metric, indicating the co-occurrence may not necessarily be a good indicator of task relatedness, at least in the context of semantic similarity.

For priority-centric metrics, we can see that our NDCG@100 had roughly 30% worse performance when compared with the median of participants' results. For both the sets of Priority F1 and Priority R scores, we performed worse still. This was somewhat expected as we did not optimise for priority in our work.

6 DISCUSSION

Comparison with other systems. Unfortunately, the performance gains from our ongoing work were not as strongly reflected in our notebook submission. However, our submitted systems achieved around the median for *Information Type F1 All* and *Information Type Accuracy* but fell short in obtaining adequate results for the *Information Type F1 Actionable*. It is clear that, despite the potential gains from using transfer learning techniques, the necessary conditions to reproduce these effects are difficult to predict prior to training. Our hypothesis that co-occurrent labels are likely to be somewhat synergistic in training did not prove to be correct.

Reflections on method. It is evident that what constitutes task synergy (for transfer) for the 2021 edition are more complex than initially expected. We believe that this method can show promising results for low-resource labels but more investigation must be carried out to better estimate the suitability of pairs of tasks and parameters prior to training.

7 CONCLUSIONS

In this paper, we proposed a method of classifying crisis and disaster documents by decomposing the multi-label problem into separate binary tasks and carrying out transfer learning between them. Further investigation into the quality of task relatedness and how to exploit the shared information between related tasks is something we have left for future work.

ACKNOWLEDGEMENTS

We would like to thank the organises of the TREC Incident Streams track for providing the datasets, evaluation metrics, and fostering a community for information retrieval in crises.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.

[2] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR*.