Method Comparison for Crisis Pipelines

Shivam Sharma
New Jersey Institute of Technology
ss4354@njit.edu

Cody Buntain
New Jersey Institute of Technology
cbuntain@njit.edu

Keywords

Incident Streams, TREC, TRECIS, crisis informatics

INTRODUCTION

In crisis informatics, data sparsity remains a crucial bottleneck for learning, and while numerous approaches exist to alleviate this issue, little domain-specific guidance exists for choosing or prioritizing these approaches. When developing a crisis informatics pipeline, there are majorly four areas to take into consideration, namely, augmentation, counter data imbalance, class selection for data augmentation, to consider which classes to augment, language model selection, and training methods. When considering these different sections of a crisis informatics pipeline, there is a lack of guidance regarding prioritization of these sections for optimization. For example, using an off-the-shelf, state-of-the-art pre-trained language model may give us some performance boost, but will an older model with better class-balanced dataset show better or similar performance, and if so, then should data augmentation be prioritized over model selection? Will a pipeline with multi-task learning give similar performance on a not so well-balanced dataset, and if so, should using better training methods be prioritized over selecting the best augmentation technique? These are some of the questions we aim to consider through our work. This paper provides this much needed domain-specific guidance by evaluating performance improvements across a series of data augmentations, model improvements, and learning designs.

Since its inception in 2018, the Incident Streams track at the annual Text Retrieval Conference(TREC-IS) has received various submissions for the information classification and priority scoring tasks. These submissions are generated from pipelines that utilize state-of-the-art language models for result generation, various augmentation techniques to counter imbalance in the data, and various other methods which might help elevate the efficiency of their pipelines. However, there is a need for a comparison between different methods that can be used within different sections of a pipeline. This work aims to fill this gap by comparing different methods in three major sections of a pipeline, namely, data augmentation, model selection, and model training.

In particular, this paper describes our research around TREC-IS and aims to answer the following research questions:

RQ1: Across several augmentation techniques and libraries, how much performance increase is observed, on average as well as for each technique or library, when applied to crisis informatics data?

RQ2: What is the change in performance when we augment high priority classes, as opposed to all classes?

RQ3: By how much might one expect performance to increase by using increasingly sophisticated off-the-shelf, pre-trained models?

RQ4: How much performance increase is observed upon using single multi-task learning pipeline, as compared to different task-specific pipelines?

RQ5: Across the various sections of a crisis informatics pipeline, namely, data augmentation, langauge model selection, and training methodology, which section should be prioritized for optimization for maximum performance boost?

To answer these question, this paper presents a systematic comparison of test results showcased by different pipelines. We outline a basic pipeline architecture and make changes in the sections we aim to compare, leaving the rest same. We define a single baseline model, which will aid us to compare across different research questions and also help

identify the section of pipeline which gives us the most improvements. This work spans TREC-IS 2021-A and 2021-B, with the results being compared on the 2021-A test dataset.

Our results show that, amongst the three different augmentation strategies used, even though there is meaning in using augmentation, there is no clear winner. However, augmenting only high priority classes under a certain threshold does seem to provide more performance boost as compared to augmenting all classes which are bellow the aforementioned threshold. Our results also showcase that using off-the-shelf pre-trained models does improve the performance by a degree in tasks, however, there is more evidence in prioritizing optimization of learning methods over selecting the best pre-trained language model.

Primary contribution of this work will be of interest to those studying crisis-informatics and methods to improve the performance of their pipelines by giving an insight into which section to prioritize for optimization. Crisis Informatics researchers in particular can benefit from the comparison of different learning methods and different language models.

RESEARCH QUESTION

In this section, we dive deeper into the various research questions and outline the methods used to answer these questions.

Augmentation Comparison

Like many other real-world data, the crisis tweets collected by TREC-IS are imbalanced in the distribution across different information as well as priority labels. This is expected as there is would be a higher number of people posting a tweet showing support or giving condolences to the victims of some crisis events as opposed to the number of people requesting aid for the same crisis. To counter this imbalance various augmentation strategies can be applied to the dataset.

In this experiment, we aim to answer the following research question:

RQ: Across several augmentation techniques and libraries, how much performance increase is observed, on average as well as for each technique or library, when applied to crisis informatics data?

We answer the above mentioned research question by observing the performance improvement across the following augmentation strategies for the textual data present in the tweets:

1. **Synonym-Augmentation:** In this method, we list all the verbs in the given tweet text and replace them with their synonyms, thus "generating" new tweet text. If a text has more than one verb in it, then we replace one verb with its respective synonyms at a time, keeping the rest of the verbs the same.

2. **Easy Data Augmentation:** Easy Data Augmentation, or EDA, is the work presented by Wei and Zou [cite paper]. This method includes four different text processing methods, namely, synonym-replacement, random insertion, random swap, and random deletion. This method is a more advanced version of our synonym-augmentation method, with synonym-replacement in any random word instead of just the verb.

3. **AugLy:** AugLy is a data augmentation library recently developed by Facebook Research. It contains over 100 different augmentation methods across multi-modalities like image, text, video, and audio. For textual data augmentation, AugLy introduces 11 different augmentation functions, of which we have used three for this experiment, namely, word splitting, similar character replacement, and "typos" simulation.

For all of these three augmentation strategies, we define a multiplication factor, which is the number of times a tweet text gets augmented. We also augment only the "actionable" classes, which means, we augment classes which has high average priority. We use DeBERTa as the core language model for all three use cases.

Class Selection for Augmentation

An important factor in evaluation for TREC-IS is evaluation scores for "actionable" information classes. These are the top six information classes that have the highest average priority score. This raises an important question of whether we would want to augment all the classes or only those classes which are "actionable", as these are the high priority classes that are more important for emergency responders.

In this experiment, we aim to answer the following research question:

RQ: What is the change in performance when we augment high priority classes, as opposed to all classes?

To answer the above research question, we compare the performance increase between two pipelines, one with augmentation on only actionable classes, and the other with augmentation for all classes. Since augmenting all the classes will keep the overall ratio of information classes the same, thus overlooking the main motive for using augmentation, we use the minimum count method used by Wang et al. 2021. In this method, Wang augments only those classes which are below a certain threshold and calculates a multiplication factor based on the difference between the threshold and the count of tweets in that class. This multiplication factor is the number of tweet text we "generate" from a single tweet. Thus, in this experiment, we compare two pipelines, one with augmentations on all the classes with class count lower than 500 and one with augmentations on "actionable" class with class count lower than 500. For this experiment, we use EDA as the base augmentation strategy and DeBERTa as the base model.

Model Selection

Model selection is another important aspect to consider when formulating a pipeline. There are various pre-trained language models available which are trained on a high amount of data, like BERT, RoBERTa, XLM, DeBERTa, etc. The state-of-the-art limits are frequently pushed with better models, which are either trained on a greater amount of data or use a denser or more refined base architecture. This raises the question of whether constantly updating our crisis informatics pipelines with these new, off-the-shelf pre-trained language model guarantee performance improvement.

Through this experiment, we aim to answer the following research question:

RQ: By how much might one expect performance to increase by using increasingly sophisticated off-the-shelf, pre-trained models?

We answer the above mentoined question by comparing three different language models, namely, BERT, RoBERTa, and DeBERTa, to check whether using state-of-the-art models like DeBERTa showcases improved results as compared to the base models like BERT and RoBERTa. In this experiment, we change the pre-trained language model in our base pipeline. We use AugLy with multiplication factor for actionable classes only to get the augmentations in the textual data.

Multi-Task Learning

Wang et al. 2021 in their work use multi-task learning, using a combined loss from priority scoring task model and information classification model. This was in turn motivated by Zhang and Yang 2017, whose work shows evidence that parameter sharing between multiple tasks is likely to enable one task to share its learned knowledge with another. Wang, in their work, defines a parameter lambda, λ, as a loss parameter which they use as a weight for adjusting the loss of priority scoring task and information labels classification to calculate the final loss. The following equation describes the calculation for the final loss function, where L_{it} is the loss for information type classification task and L_{pri} is the loss for priority scoring task.

$$L_{total} = \lambda L_{it} + (1 - \lambda) L_{pri}$$

Through this experiment we aim to answer the following research question:

RQ: How much performance increase is observed upon using single multi-task learning pipeline, as compared to different task-specific pipelines?

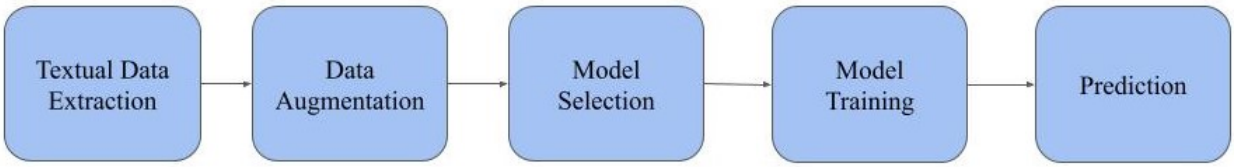

Figure 1. Base Pipeline Architecture

To answer this, we compare two different pipelines, one with multi-task learning, and one with two task-specific models, one for each, information type classification task and priority scoring task. We can get a model for priority scoring task only and information classification task only, if we set λ as 0 and 1, respectively. We use the λ parameter to train two different models, one for each task, and combine their results to form the final result. We compare this result with the multi-task learning using the equation described above with λ as 0.5. We use AugLy with multiplication factor for actionable classes only to get the augmentations in the textual data and use DeBERTa as the base neural language model.

METHODOLOGY

This sections defines the method we used to evaluate the four research questions discussed above. We outline the basic architecture of the pipeline, modifying different sections of which would generate the pipelines which we use to answer the research questions. We also discuss in brief the dataset used for training and testing, and also the evaluations metrics used.

Data Description

Since its inception in 2018, TREC-IS has accumulated a total labeled corpus of more than 60k labeled crisis tweets made during 75 different crisis events. These labeled tweets can be distributed into 5 different subsets, based on their editions, namely, 2018, 2019-A, 2019-B, 2020-A, and 2020-B. For our experiments in this work, we use the whole dataset for training and use the first labeled dataset released for the 2021 edition as the testing dataset.

Evaluation Metrics

TREC-IS releases an evaluation notebook for every edition. These notebooks include, an nDCG metric for ranking content by priority, F1 score divided into two sets, one restricted to "actionable" information types, and the other containing all possible labels, and R score for priority score comparison which, similar to the F1 score, is also divided into two sets, one for "actionable" information type and one for all possible labels. This "actionable" set is restricted to the top six information types with the highest average priority score, making them the most "important" types to classify correctly in a qualitative sense.

Model Description

We keep a consistent model architecture across all the experiments described in this work. This works aim to compare the results between three different sections of a neural pipeline, namely, augmentation, pre-trained model, and training method. We change the specific sections of the basic neural pipeline architecture and keep everything else the same for a clear and consistent comparison. Our model architecture is heavily inspired by the model architecture used by the model described by Wang et al. 2021. Figure 1 describes our basic pipeline architecture, which can be split into five major sections, namely, textual data extraction, data augmentation, model selection, model training, and prediction. Of these five, in this work, we change the data augmentation, the model selection, and the model training sections for their respective experiments.

RESULTS

This section presents the results for the various experiments outlined above. We compare the various pipelines based on the evaluation scores as well as across individual information type scores. We also the compare the net percent improvement from our baseline.

Table 1. Evaluation results on test dataset by pipelines with different augmentation techniques.

Augmentation Statergies	nDCG@100	Info Type F1		Info Accuracy	Priority F1		Priority R	
		Actionable	All		Actionable	All	Actionable	All
Baseline	0.492	0.2062	0.2837	0.8896	0.215	0.1737	0.0643	0.155
Augly	**0.5112**	0.2132	0.2927	0.8883	**0.257**	0.2012	**0.1661**	**0.2126**
EDA	0.4538	**0.3008**	**0.3239**	0.8817	0.2434	0.1855	0.1046	0.1881
Synonym Augmentation	0.4831	0.285	0.3189	**0.891**	0.2507	**0.2045**	0.1342	0.1924

Table 2. Per-information label score distribution for various augmentation pipelines

Information Classes	Precision				Recall				F1 Score			
	Baseline	EDA	AugLy	Syn-Aug	Baseline	EDA	AugLy	Syn-Aug	Baseline	EDA	AugLy	Syn-Aug
Actionable Classes												
EmergingThreats	0.14	0.14	0.15	**0.16**	0.3	**0.36**	0.24	0.26	0.19	**0.2**	0.18	**0.2**
GoodsServices	0	0.61	0.61	**0.65**	0	**0.38**	0.13	0.26	0	**0.46**	0.22	0.37
MovePeople	**0.63**	0.5	0.51	0.56	0.32	**0.53**	0.27	0.4	0.43	**0.52**	0.35	0.47
NewSubEvent	0.12	0.02	**0.15**	0.09	**0.14**	0.04	0.1	0.06	**0.13**	0.02	0.12	0.07
SearchAndRescue	**0.22**	0.13	0.04	0.19	0.06	0.21	0.03	**0.24**	0.09	0.16	0.03	**0.21**
ServiceAvailable	0.46	0.43	0.44	**0.51**	0.34	**0.45**	0.32	0.32	0.39	**0.44**	0.37	0.39
Non-Actionable Classes												
Advice	**0.51**	0.39	0.44	0.46	0.24	0.29	0.23	**0.37**	0.33	0.33	0.31	**0.41**
CleanUp	0.19	0.25	0.23	**0.27**	0.5	**0.58**	0.54	0.54	0.28	0.35	0.33	**0.36**
ContextualInformation	**0.12**	0.09	0.11	0.11	**0.11**	0.04	0.07	0.07	**0.11**	0.06	0.09	0.09
Discussion	**0.03**	**0.03**	0.02	0.02	0.1	**0.17**	0.15	0.15	**0.05**	**0.05**	0.04	0.04
Donations	0.32	**0.44**	0.3	0.4	0.56	0.65	0.57	**0.68**	0.41	**0.53**	0.4	0.5
Factoid	**0.54**	0.45	0.48	0.5	0.49	**0.56**	0.49	0.5	**0.52**	0.5	0.49	0.5
FirstPartyObservation	0.14	0.13	0.17	0.16	0.18	0.17	0.27	**0.32**	0.16	0.14	**0.21**	**0.21**
Hashtags	**0.47**	0.38	0.41	0.39	0.49	0.55	**0.56**	0.46	**0.48**	0.45	0.47	0.42
InformationWanted	0.4	0.49	0.45	**0.5**	0.18	0.49	0.39	**0.51**	0.25	0.49	0.42	**0.5**
Irrelevant	0.62	0.71	0.68	**0.73**	0.47	0.42	0.47	**0.48**	0.53	0.43	0.55	**0.58**
Location	**0.6**	0.56	0.59	0.59	0.66	**0.7**	0.61	0.63	**0.63**	0.62	0.6	0.61
MultimediaShare	0.28	0.3	0.31	**0.31**	0.4	0.46	**0.51**	0.44	0.33	0.36	**0.38**	0.36
News	**0.27**	0.26	0.26	0.25	0.33	**0.36**	0.24	0.16	**0.3**	**0.3**	0.25	0.2
Official	0.11	0.13	**0.16**	0.11	0.11	**0.12**	0.04	0.03	0.11	**0.12**	0.07	0.05
OriginalEvent	0.03	0.04	**0.06**	0.04	0	**0.01**	**0.01**	0	0.01	0.01	**0.02**	0.01
Sentiment	0.33	0.33	0.33	**0.35**	0.36	0.35	0.41	**0.45**	0.35	0.34	0.37	**0.39**
ThirdPartyObservation	0.49	0.45	**0.5**	0.48	0.26	0.26	0.29	0.27	0.34	0.33	**0.37**	0.35
Volunteer	**0.25**	0.2	0.17	0.16	0.06	**0.35**	0.23	0.25	0.1	**0.25**	0.2	0.19
Weather	**0.7**	0.65	0.66	0.68	**0.49**	0.46	0.4	0.39	**0.58**	0.54	0.49	0.49

Augmentation Comparison

Table 1 shows the results from the various augmentation methods. As evident, there is clearly no "best" augmentation strategy. AugLy outperforms the rest of the strategies in all of the priority scores, except for priority F1 for all, where it is still competitive to the best and the difference can be attributed to the randomness. EDA, on the other hand, outperforms the other strategies in information type classification scores, except for the information type accuracy, which is competitive to Synonym-Augmentation and the difference is low enough that it can be attributed to randomness.

Table 2 shows the results for individual information label for the three augmentation pipelines and also the baseline. For the F1 score for actionable classes, EDA pipeline outperforms the other pipelines in majority classes and apart from the poor performance in "NewSubEvent", it is competitive to the best score in "SearchAndRescue" label. However, for non-actionable classes, pipeline with synonym-augmentation shows better results than pipeline using EDA.

Class Selection for Augmentation

This section discusses the results for the comparison between augmentation on actionable classes only and on all classes. The "Actionable" pipeline is the pipeline with augmentations done on classes which are actionable as well

Table 3. Test Result comparison between pipeline augmenting only "Actionable" information classes and pipeline augmenting "All" information classes.

Augmentation Statergies	nDCG@100	Info Type F1		Info Accuracy	Priority F1		Priority R	
		Actionable	All		Actionable	All	Actionable	All
Baseline	**0.492**	0.2062	0.2837	0.8896	0.215	0.1737	0.0643	0.155
Actionable	0.4837	**0.2988**	**0.3305**	0.889	**0.2437**	0.191	**0.1396**	**0.2087**
All	0.4915	0.2754	0.3147	**0.8897**	0.2352	**0.1922**	0.1242	**0.2087**

Table 4. Per-information label score distribution for class selection pipelines.

Information Classes	Precision			Recall			F1-Score		
	Baseline	Actionable	All	Baseline	Actionable	All	Baseline	Actionable	All
Actionable Classes									
EmergingThreats	0.14	0.14	**0.16**	**0.3**	0.24	0.26	**0.19**	0.18	0.2
GoodsServices	0	0.42	**0.58**	0	**0.35**	0.19	0	**0.38**	0.28
MovePeople	**0.63**	0.57	0.56	0.32	**0.42**	0.37	0.43	**0.48**	0.45
NewSubEvent	0.12	**0.14**	0.11	0.14	**0.15**	0.12	0.13	**0.14**	0.12
SearchAndRescue	**0.22**	0.15	0.11	0.06	**0.41**	0.21	0.09	**0.22**	0.14
ServiceAvailable	0.46	0.5	**0.55**	0.34	0.32	**0.4**	0.39	0.39	**0.46**
Non-Actionable Classes									
Advice	**0.51**	0.45	0.49	0.24	0.31	**0.35**	0.33	0.37	**0.4**
CleanUp	0.19	**0.21**	0.1	0.5	0.54	**0.58**	0.28	**0.31**	0.17
ContextualInformation	**0.12**	0.07	0.1	**0.11**	0.05	0.05	**0.11**	0.06	0.07
Discussion	**0.03**	**0.03**	**0.03**	0.1	**0.2**	0.15	**0.05**	**0.05**	**0.05**
Donations	0.32	0.31	**0.35**	0.56	**0.62**	0.59	0.41	0.42	**0.44**
Factoid	**0.54**	0.51	0.5	0.49	0.54	**0.55**	0.52	**0.53**	0.52
FirstPartyObservation	0.14	**0.16**	0.14	0.18	**0.26**	0.22	0.16	**0.2**	0.17
Hashtags	**0.47**	0.41	0.43	0.49	**0.63**	0.6	0.48	**0.5**	**0.5**
InformationWanted	0.4	**0.47**	0.46	0.18	0.52	**0.66**	0.25	0.49	**0.54**
Irrelevant	0.62	0.75	**0.76**	**0.47**	0.43	0.41	0.53	**0.54**	0.53
Location	**0.6**	0.58	0.58	**0.66**	**0.66**	0.59	**0.63**	0.62	0.58
MultimediaShare	0.28	**0.32**	0.29	0.4	**0.54**	0.43	0.33	**0.4**	0.35
News	0.27	**0.28**	**0.28**	**0.33**	0.24	0.27	**0.3**	0.26	0.28
Official	0.11	**0.15**	0.12	**0.11**	0.05	0.06	**0.11**	0.08	0.08
OriginalEvent	0.03	**0.07**	0.06	0	**0.01**	**0.01**	0.01	**0.02**	0.01
Sentiment	0.33	**0.36**	0.35	0.36	**0.45**	0.42	0.35	**0.4**	0.39
ThirdPartyObservation	**0.49**	0.47	**0.49**	0.26	**0.36**	0.34	0.34	**0.41**	0.4
Volunteer	**0.25**	**0.25**	0.17	0.06	**0.33**	0.4	0.1	**0.29**	0.24
Weather	**0.7**	0.66	0.66	**0.49**	0.46	0.39	**0.58**	0.54	0.49

Table 5. Test results comparision between pipelines using different language models.

Augmentation Statergies	nDCG@100	Info Type F1		Info Accuracy	Priority F1		Priority R	
		Actionable	All		Actionable	All	Actionable	All
Baseline	0.492	0.2062	0.2837	0.8896	0.215	0.1737	0.0643	0.155
BERT	0.4585	0.1617	0.2382	0.8872	0.2393	0.2004	0.1098	0.1522
RoBERTa	0.4987	**0.2749**	**0.3153**	**0.8926**	0.2238	0.1835	**0.1966**	0.2074
DeBERTa	**0.5112**	0.2132	0.2927	0.8883	**0.257**	**0.2012**	0.1661	**0.2126**

as bellow a threshold of 500 tweet text, per information class. The "All" pipeline is the pipeline with augmentations done on all the classes which are bellow the threshold of 500 tweets text, per class, irrespective of whether or not those classes are actionable or not.

As evident from the table 3, the "Actionable" pipeline outperforms in all of the major evaluation scores, and is competitive in the rest. The two pipelines are outperformed by the Baseline in the nDCG score, but the difference is close enough that this may be attributed to the randomness of the model. The "All" pipeline outperforms the rest in information type accuracy score and the Priority F1 for all classes, but the difference between the "Actionable" and "All" pipelines for these score is near enough that this may also be attributed to randomness of the model.

Table 4 showcases evaluation scores for individual information type labels. As evident from the table, for the F1 score for actionable classes, the "Actionable" pipeline outperforms the "All" pipeline in majority of the classes, with an exception for "EmergingThreats" and "ServiceAvailable". We can observe similar results for the F1 score of non-actionable classes as well, with the "Actionable" pipeline outperforming the "All" pipeline in more than 50% of the information classes.

Model Selection

Table 5 compares the evaluation results for the three pre-trained model runs, namely, BERT, RoBERTa, and DeBERTa. As evident from the table, RoBERTa outperforms the rest in the information type classification task but shows competitive results in the priority scoring task, with the best scores for Priority R for actionable classes. Similarly, DeBERTa outperformed the rest in the priority scoring task, except for Priority R for actionable classes. It is important to note that the Baseline pipeline is performing better than the BERT pipeline, and shows competitive results against the DeBERTa model, in all three evaluation scores information label classification task.

Table 6 compares the scores from individual information type labels. For F1 scores for Actionable classes, RoBERTa outperforms the rest of the pipelines in three of the six information classes. For the F1 score for non-actionable classes as well, RoBERTa outperforms the rest of the pipelines in more than 50% of the information labels. Through

Table 6. Per-information label score distribution for model selection pipelines.

Information Classes	Precision				Recall				F1 Score			
	Baseline	BERT	RoBERTa	DeBERTa	Baseline	BERT	RoBERTa	DeBERTa	Baseline	BERT	RoBERTa	DeBERTa
Actionable Classes												
EmergingThreats	0.14	0.11	0.13	**0.15**	**0.3**	0.17	0.26	0.24	**0.19**	0.13	0.17	0.18
GoodsServices	0	0.56	**0.66**	0.61	0	0.05	**0.3**	0.13	0	0.1	**0.41**	0.22
MovePeople	**0.63**	0.42	0.54	0.51	0.32	0.14	0.42	0.27	0.43	0.21	**0.47**	0.35
NewSubEvent	0.12	0.09	0.07	**0.15**	**0.14**	0.06	0.05	0.1	**0.13**	0.08	0.06	0.12
SearchAndRescue	**0.22**	0.18	0.09	0.04	0.06	**0.09**	**0.09**	0.03	0.09	**0.12**	0.09	0.03
ServiceAvailable	0.46	0.48	**0.54**	0.44	0.34	0.26	**0.38**	0.32	0.39	0.34	**0.45**	0.37
Non-Actionable Classes												
Advice	**0.51**	0.47	**0.51**	0.44	0.24	0.23	**0.3**	0.23	0.33	0.31	**0.38**	0.31
CleanUp	0.19	0.23	**0.25**	0.23	0.5	0.42	0.5	**0.54**	0.28	0.3	**0.33**	**0.33**
ContextualInformation	0.12	**0.18**	0.14	0.11	0.11	**0.12**	0.06	0.07	0.11	**0.14**	0.09	0.09
Discussion	**0.03**	0.01	0.02	0.02	0.1	0.05	0.1	**0.15**	**0.05**	0.02	0.04	0.04
Donations	**0.32**	0.31	0.3	0.3	0.56	0.37	**0.61**	0.57	**0.41**	0.34	0.4	0.4
Factoid	0.54	0.51	**0.55**	0.48	0.49	0.46	**0.58**	0.49	0.52	0.48	**0.56**	0.49
FirstPartyObservation	0.14	0.12	**0.19**	0.17	0.18	0.13	0.26	**0.27**	0.16	0.12	**0.22**	0.21
Hashtags	0.47	**0.51**	0.42	0.41	0.49	0.42	**0.62**	0.56	0.48	0.46	**0.5**	0.47
InformationWanted	0.4	0.11	**0.59**	0.45	0.18	0.01	0.3	**0.39**	0.25	0.02	0.4	**0.42**
Irrelevant	0.62	0.5	**0.71**	0.68	0.47	**0.49**	0.48	0.47	0.53	0.5	**0.58**	0.55
Location	0.6	0.6	**0.62**	0.59	**0.66**	0.59	0.59	0.61	**0.63**	0.59	0.61	0.6
MultimediaShare	0.28	0.26	**0.34**	0.31	0.4	0.34	**0.52**	0.51	0.33	0.3	**0.41**	0.38
News	**0.27**	0.25	0.25	0.26	**0.33**	0.29	0.23	0.24	**0.3**	0.27	0.24	0.25
Official	0.11	0.11	**0.26**	0.16	**0.11**	0.06	0.07	0.04	**0.11**	0.08	0.1	0.07
OriginalEvent	0.03	0.04	0.04	**0.06**	0	0	**0.01**	**0.01**	0.01	0.01	**0.02**	**0.02**
Sentiment	0.33	0.31	**0.35**	0.33	0.36	0.34	**0.47**	0.41	0.35	0.32	**0.4**	0.37
ThirdPartyObservation	0.49	0.48	**0.54**	0.5	0.26	0.23	0.28	**0.29**	0.34	0.31	**0.37**	**0.37**
Volunteer	**0.25**	0	**0.25**	0.17	0.06	0	0.04	**0.23**	0.1	0	0.07	**0.2**
Weather	**0.7**	0.6	0.66	0.66	**0.49**	0.33	0.43	0.4	**0.58**	0.43	0.52	0.49

Table 7. Test result comparison between pipeline using multi-task learning and pipeline using two different task-specific models.

Augmentation Statergies	nDCG@100	Info Type F1		Info Accuracy	Priority F1		Priority R	
		Actionable	All		Actionable	All	Actionable	All
Baseline	0.492	0.2062	0.2837	**0.8896**	0.215	0.1737	0.0643	0.155
Separate Task	0.428	**0.2256**	0.2895	0.8876	0.1696	0.1336	0.1053	0.1875
Multi-Task	**0.5112**	0.2132	**0.2927**	0.8883	**0.257**	**0.2012**	**0.1661**	**0.2126**

this table, we can also observe the competitive results shown by the Baseline pipeline over the BERT and the DeBERTa pipelines.

Multi-Task Learning

This section shows the results obtained for the experiment between a single multi-task pipeline, and two separate task-specific pipelines, one for each information label classification task and as well as one for priority scoring task. In the tables 7 and 8, we discuss these two pipelines, with “Separate Task" representing the pipeline using two separate task-specific models, and “Multi-Task” representing the multi-task learning pipeline.

As evident from table 7, the multi-task learning outperforms the separate task pipelines in almost all of the evaluation scores, with an exception for actionable information type F1 score and the information type accuracy. As compared to the Baseline, the “Multi-Task” pipeline has a noticeable improvement in most of the scores, with an exception for information type accuracy. For the evaluation scores regarding information label classification task, we observe that the results by “Separate Task” pipeline is competitive to the results showcased by the “Multi-Task” pipeline, however, multi-task learning outperforms the separate task pipeline by a wide margin.

Table 8 shows the evaluation results for individual information labels for the two mentioned pipelines. As evident from the table, the “Multi-Task” pipeline does not show a significant performance boost as compared to the “Separate Task” pipeline. The “Multi-Task” pipeline performs poorly for the actionable information classes, and showcases the best results in less than half of the non-actionable information classes.

Table 8. Per-information label score distribution for different learning methods pipelines.

Information Classes	Precision			Recall			F1 Score		
	Baseline	**Separate Tasks**	**Multi-Task**	**Baseline**	**Separate Tasks**	**Multi-Task**	**Baseline**	**Separate Tasks**	**Multi-Task**
Actionable Classes									
EmergingThreats	0.14	0.13	**0.15**	**0.3**	0.21	0.24	**0.19**	0.16	0.18
GoodsServices	0	0.58	**0.61**	0	**0.16**	0.13	0	**0.25**	0.22
MovePeople	**0.63**	0.54	0.51	**0.32**	0.29	0.27	**0.43**	0.38	0.35
NewSubEvent	0.12	0.06	**0.15**	**0.14**	0.05	0.1	**0.13**	0.06	0.12
SearchAndRescue	**0.22**	0.12	0.04	0.06	**0.09**	0.03	0.09	**0.1**	0.03
ServiceAvailable	**0.46**	**0.46**	0.44	0.34	**0.38**	0.32	0.39	**0.41**	0.37
Non-Actionable Classes									
Advice	**0.51**	0.4	0.44	**0.24**	0.2	0.23	**0.33**	0.27	0.31
CleanUp	0.19	**0.28**	0.23	0.5	**0.54**	**0.54**	0.28	**0.37**	0.33
ContextualInformation	**0.12**	0.09	0.11	**0.11**	0.08	0.07	**0.11**	0.09	0.09
Discussion	**0.03**	**0.03**	0.02	0.1	**0.15**	**0.15**	**0.05**	**0.05**	0.04
Donations	0.32	**0.37**	0.3	0.56	**0.58**	0.57	0.41	**0.46**	0.4
Factoid	**0.54**	0.5	0.48	0.49	**0.5**	0.49	**0.52**	0.5	0.49
FirstPartyObservation	0.14	0.14	**0.17**	0.18	0.26	**0.27**	0.16	0.18	**0.21**
Hashtags	**0.47**	0.4	0.41	0.49	0.54	**0.56**	**0.48**	0.46	0.47
InformationWanted	0.4	**0.45**	**0.45**	0.18	0.32	**0.39**	0.25	0.37	**0.42**
Irrelevant	0.62	**0.69**	0.68	**0.47**	0.45	0.47	0.53	0.54	**0.55**
Location	0.6	**0.61**	0.59	**0.66**	0.55	0.61	**0.63**	0.57	0.6
MultimediaShare	0.28	**0.31**	**0.31**	0.4	0.46	**0.51**	0.33	0.37	**0.38**
News	**0.27**	0.25	0.26	**0.33**	0.26	0.24	**0.3**	0.26	0.25
Official	0.11	**0.16**	**0.16**	**0.11**	0.05	0.04	**0.11**	0.08	0.07
OriginalEvent	0.03	**0.06**	**0.06**	0	**0.01**	**0.01**	0.01	**0.02**	**0.02**
Sentiment	**0.33**	0.3	0.33	0.36	0.4	**0.41**	0.35	0.34	**0.37**
ThirdPartyObservation	0.49	**0.52**	0.5	0.26	**0.29**	0.29	0.34	**0.37**	**0.37**
Volunteer	**0.25**	0.13	0.17	0.06	0.15	**0.23**	0.1	0.14	**0.2**
Weather	**0.7**	0.65	0.66	**0.49**	0.33	0.4	**0.58**	0.44	0.49

REFERENCES

Wang, C., Nulty, P., and Lillis, D. (2021). "Transformer-based Multi-task Learning for Disaster Tweet Categorisation". In: *Proceedings of the International ISCRAM Conference* 2021-May.May.

Zhang, Y. and Yang, Q. (2017). "A survey on multi-task learning". In: *arXiv preprint arXiv:1707.08114*.