IRIT-IRIS at TREC 2022: CrisisFACTS Track

Alexis Dusart, Gilles Hubert, and Karen Pinel-Sauvagnat

{alexis.dusart, gilles.hubert, karen.sauvagnat}@irit.fr,
IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3
118 route de Narbonne, F-31062 Toulouse cedex 9, France

Abstract. This paper presents the approaches proposed by the IRIS team of the IRIT laboratory for the TREC CrisisFACTS track. The CrisisFACTS track aims to summarize online data sources during crisis events. In our participation, we used neural language models according to three different strategies.

1 Introduction

In 2022, we participated in the CrisisFACTS track. In this notebook, we first briefly present the track in section 2, then we present our proposed approaches for the track in section 3, and the runs associated to these approaches in section 4. Finally, we report preliminary results in section 5.

2 Overview of the CrisisFacts track

The purpose of the TREC CirsisFACTS track is to provide a summary of information from online sources during crisis events, as shown in Figure 1. The summary should be built incrementally using all the information available since the beginning of the event. As explained by the organizers, a summary might be generated by an emergency response staff at the start of a new shift to inform themselves of new developments[1]. The sources of information used in this track are the news, as well as Twitter, Facebook, and Reddit data, split in itemised text snippets. The participant systems of the 2022 task have to list minimally-redundant important facts (items in case of extractive approach) with their score of importance denoting how critical the fact is for responders.

3 Proposed approaches

For all approaches we want to take advantage of neural language models (NLM). As streams are too long to encode the entire text with such models, we used two strategies: (i) the first using NLM to encode an item and the frequency of the stream terms, that we called TSSuBERT, and (ii) the second using NLM to encode each item, then computing the representation of the stream as the mean representation of each item that composes it, that we called DAN-TSS. In the next paragraphs we detail the approaches.

`TSSuBERT`: For this approach, the system computes the importance using a neural pre-trained language model and the frequency of the stream tokens. The Figure 2 illustrates this importance prediction part with a tweet. Then, the system selects items to keep in the summary using

[1] `https://crisisfacts.github.io/`

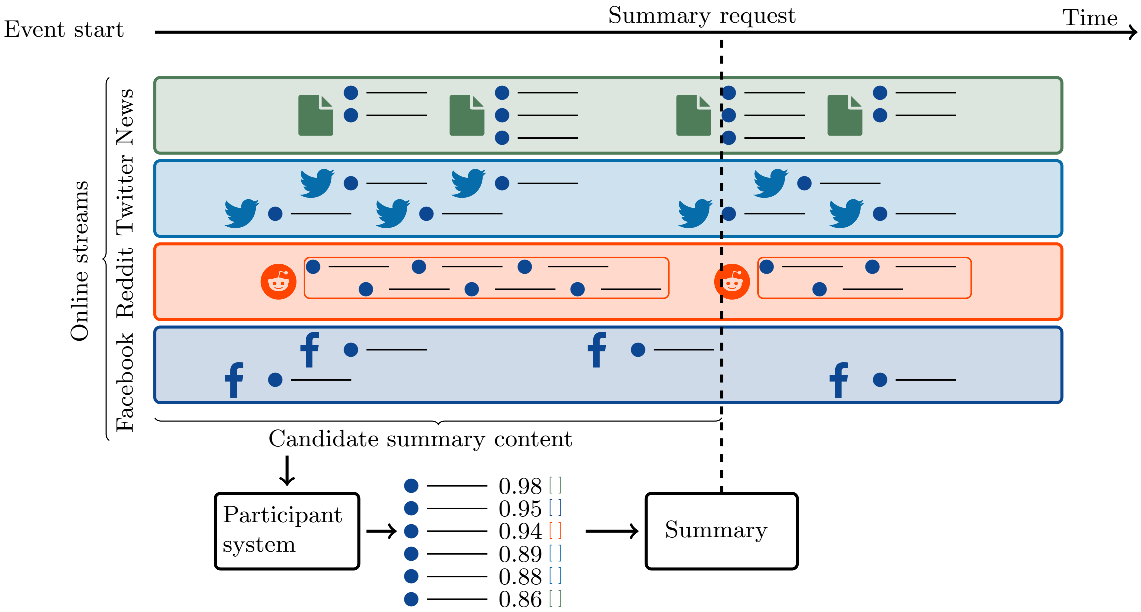

Fig. 1. CrisisFACTS track task, as shown on the track's website (`https://crisisfacts.github.io/`)

redundancy removal in the manner of MMR. For each candidate item, regarding the importance score, a similarity score is computed between the item and the items already kept for the summary. If the similarity score is lower than a similarity threshold between the item and each item already kept for the summary, the item is kept for the summary.

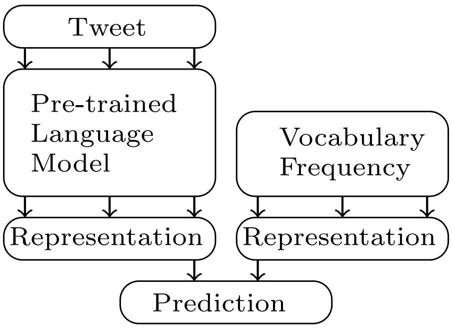

Fig. 2. Architecture of the importance prediction part.

`DAN-TSS`: For this approach, the system computes the importance score as the similarity between an item and a potential Oracle summary automatically constructed. The potential Oracle summary is first created using the mean of all the items of the daily stream, regarding the item representations. Then, using the mean representation, the representation of a potential Oracle summary is generated with a trained model based on the Deep Average Network (DAN) model [1]. To limit redundancy, once the first item is retrieved for a day (e.g., the item with the highest score), an item is represented as the mean of all the items already retrieved plus it. The importance score of an item is then computed as the similarity between the item representation and the potential Oracle summary multiplied by the importance score of the previous item kept for the day.

4 Runs

Using the approaches presented in section 3, we submitted three runs named IRIT_IRIS_tssubert, IRIT_IRIS_mean_USE, and IRIT_IRIS_mean_USE_InformationNeeds. All these runs are extractive and automatic. One run, the IRIT_IRIS_mean_USE_InformationNeeds run uses the list of general and disaster-specific queries while the others do not use them. In the next paragraphs we detail the runs.

IRIT_IRIS_tssubert: This run is based on the TSSuBERT approach. We used DistilBERT [2] as neural language model. The similarity score is computed as Cosine similarity. At last, the similarity threshold was set as follows, with length expressed in number of tokens:

- $\mathit{similarity} = 0.3$ if $\mathit{length}(\mathit{summary}) < 50$
- $\mathit{similarity} = 0.3 * \frac{\log(50)}{\log(\mathit{length}(\mathit{summary}))}$ else

This adaptive threshold aims at avoiding redundancy in summaries and at reducing the size of the predicted summaries. We trained the prediction score model on the ISSumSet dataset [3], except the events appearing in the track events. More implementation details can be found in [4].

IRIT_IRIS_mean_USE: This run is based on the DAN-TSS approach. We used USE (Universal Sentence Encoder) [5] to generate item representations. We trained this model on the ISSumSet dataset [3], except the events appearing in the track events. The similarity score is computed as Cosine similarity.

IRIT_IRIS_mean_USE_InformationNeeds: This run is also based on the DAN-TSS approach. This run is almost the same as the previous run (IRIT_IRIS_mean_USE). However, this run uses the list of general and disaster-specific queries. Unlike the IRIT_IRIS_mean_USE run, the potential Oracle summary is created by first using the mean of all the items of the daily stream retrieved by the pyTerrier [6] retriever with the *DFReeKLIM* model. Using this mean representation, a potential Oracle summary is then generate with the trained model. The importance score of an item is then computed as many times as the number of queries, and we considered the highest of these scores as the score of the item.

As suggested by the organizers, we returned 100 (the 100 highest scored) facts per summary request for the runs IRIT_IRIS_mean_USE_InformationNeeds and IRIT_IRIS_mean_USE. To reach at least 100 facts per summary request for the run IRIT_IRIS_mean_USE_InformationNeeds we returned up to the 6 higest scored facts per query.

5 Results

We report in Table 1 a summary of the automatic evaluation regarding our runs and the average of Min, Median, and Max concerning all the runs. We can see that our runs are around the median

(9 higher the median, and 9 lower). We can see that each run is of interest regarding the different evaluations: (i) `IRIT_IRIS_mean_USE` is better regarding the NIST summaries and BertScore F1 using wiki summaries, (ii) `IRIT_IRIS_mean_USE_INeeds` is better regarding ics summaries and ROUGE2 F1 using NIST summaries, and (iii) `IRIT_IRIS_tssubert` is better regarding ROUGE2 F1 using wiki summaries. In light of these results, an in-depth analysis of the results regarding the automatic evaluation as well as the manual assessment will allow us to identify the strengths and limitations of our approaches.

Run	ICS		NIST		Wiki	
	BertScore F1	ROUGE2 F1	BertScore F1	ROUGE2 F1	BertScore F1	ROUGE2 F1
Min (Average)	0.415	0.011	0.502	0.049	0.471	0.018
Median (Average)	0.440	0.040	0.546	0.122	0.520	0.027
Max (Average)	0.466	0.059	0.590	0.163	0.575	0.041
IRIT_IRIS_mean_USE	0.439	0.034	0.554	0.126	0.533	0.029
IRIT_IRIS_mean_USE_INeeds	0.455	0.037	0.552	0.126	0.518	0.024
IRIT_IRIS_tssubert	0.430	0.014	0.551	0.065	0.507	0.032

Table 1. Summary of automatic evaluation of runs `IRIT_IRIS_mean_USE`, `IRIT_IRIS_mean_USE_INeeds` and `IRIT_IRIS_tssubert` as well as average of Min, Median, and Max results.

6 Conclusion

In this paper, we presented our approaches for the TREC CrisisFACTS 2022 track, which aims at summarizing online data in order to help emergency services in case of crisis events. We proposed approaches based on neural language models. The evaluation results are globally encouraging. A more in-depth analysis, will allow us to identify the strengths and limitations of the proposed approaches.

References

1. Mohit Iyyer, Varun Manjunatha, Jordan L. Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1681–1691. The Association for Computer Linguistics, 2015.
2. Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.
3. Alexis Dusart, Karen Pinel-Sauvagnat, and Gilles Hubert. Issumset: a tweet summarization dataset hidden in a TREC track. In *SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, Republic of Korea, March 22-26, 2021*, pages 665–671. ACM, 2021.
4. Alexis Dusart, Karen Pinel-Sauvagnat, and Gilles Hubert. Tssubert: Tweet stream summarization using BERT. *CoRR*, abs/2106.08770, 2021.
5. Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal sentence encoder for english. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on*

Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018, pages 169–174. Association for Computational Linguistics, 2018.

6. Craig Macdonald and Nicola Tonellotto. Declarative experimentation ininformation retrieval using pyterrier. In *Proceedings of ICTIR 2020*, 2020.