L3S at the TREC 2022 CrisisFACTS Track

Thi Huyen Nguyen[1] and Koustav Rudra[2]

[1] L3S Research Center, Hannover, Germany
nguyen@l3s.de
[2] Indian Institute of Technology (Indian School of Mines) Dhanbad, India
koustav@iitism.ac.in

Abstract. This paper describes our proposed approach for the multi-stream summarization of the crisis-related events in the TREC 2022 CrisisFACTS track [2]. We apply a retrieval and ranking-based two-step summarization approach. First, we employ a sparse retrieval framework where content texts from multiple online streams are treated as a document corpus, and a term matching-based retrieval strategy is used to retrieve relevant contents, so-called facts, to the set of queries in a given event day. Next, we use several pre-trained models to measure semantic similarity between query-fact or fact-fact pairs, score and rank the facts for the extraction of daily event summaries.

Keywords: Crisis events · summarization · retrieval

1 Introduction

Summarization has been a topic of increasing interest in recent years. During crises, it is important to generate short, informative, and non-redundant summaries of the events to help local communities and stakeholders obtain prompt updates and react accordingly. Many recent works have proposed various summarization approaches [5, 8–10, 12]. However, these studies assume that all the input texts of the summarization models are relevant to the event. Some of them filter out irrelevant content by applying classification methods, yet it requires labeled data for training the classifiers. Besides, previous studies mainly focus on specific traits of a certain data source, such as Twitter or news articles, but not multi-stream data for summarization.

This work focuses on summarization of multi-stream datasets that are suitable to the information needs of emergency responders. We take advantage of both sparse and dense representation techniques to find the most relevant, important, and diverse facts as event summaries. More specifically, we apply a sparse retrieval framework that allows quick indexing and retrieval of facts for a given query set. However, this step only relies on simple term-based matching and does not capture the semantic matching among queries and facts. Hence, we further employ multiple language models pre-trained on various sentence similarity tasks (details Section 2) to extract semantic representations of queries and facts for ranking and extracting daywise event summaries. Earlier, Akula et al [3]

showed that a combination of different semantic representation approaches gives effective results in summarization. Hence, we use an ensemble of different semantic similarity scores to get the final scores of facts obtained in the term-based matching stage. In this paper, we propose two different ranking approaches to retrieve the event summary. First, we rely on the semantic similarity scores between query-fact pairs and retrieve facts that have the highest mean similarity scores with queries as the summary. In the second approach, we only consider fact-fact pairs to form a graph based on the semantic similarity among the facts and retrieve the final summary.

2 Approach

Given a list of content text $T = \{t_1, t_2, .., t_n\}$ from multiple streams (i.e., Twitter, Reddit, Web News) and a set of queries $Q = \{q_1, q_2, .., q_k\}$ defining the information needs of stakeholders in a specific event day, our aim is to return a list of maximum K relevant content texts, so-called facts ($S \subset T$), along with their importance scores as our daywise system summary. In this section, we introduce our proposed summarization approach, which consists of two stages — (i). Relevant Fact Retrieval, (ii). Ranking and Summarization. We elaborate our retrieval and two different ranking approaches to extract informative and diverse daywise summaries of disaster events in the following parts.

2.1 Fact Retrieval

During crisis events, human organizations or stakeholders want to receive a summary with respect to specific information needs. However, CrisisFACTS dataset contains information from different sources such as Twitter, Reddit, Facebook, etc., and many contents do not satisfy the specific requirements mentioned in queries. Side by side, the collection of content texts is significantly large, which may hamper the fast functioning of the summarization step (Section 2.2). Hence, we first apply a term-based matching approach to gather the relevant content (so-called facts) for a given query set. We use pyTerrier [6], which is a Python retrieval framework. It provides the implementation of several term-based matching models, but we choose to use DFReeKLIM [4] due to its effectiveness in short document retrieval tasks [7]. Given a set of texts T from multiple streams and a list of queries Q in each event day, we obtain a subset of facts $RF \subset T$, which consists of relevant content texts to the query set Q. These facts are then used as input to our summarization methods described next.

2.2 Ranking and Summarization

In this section, we propose two different summarization approaches based on facts retrieved in the first phase.

Query-based Summarization DFreeKLIM model retrieves a set of relevant content with respect to each query in the query set (Q) from the entire text corpus (T). Hence, we obtained a set of facts for each of the queries. Let's say, for query q, the set of facts is presented via set RF_q. However, semantic aspects are missing in this step.

First, we calculate the semantic score between queries $q \in Q$ and all the facts $f \in RF, RF = \cup_{q \in Q} RF_q$. To compute the score, we extract contextual embeddings of facts and queries using multiple language models that were pre-trained on several sentence pair tasks, such as semantic text similarity, paraphrase detection, natural language inference, and passage retrieval. Given M pre-trained models, we compute M cosine similarity scores between embeddings of a fact and a query. The final similarity score between a fact and a query is the average of M similarity scores. Our aim is to extract the most relevant and important facts with respect to queries of the event day as a summary. The importance score of a fact is the average similarity score between the fact and all the queries. Figure 1 illustrates steps to compute importance score of facts in an event day.

We select facts gradually and in chronological order for our final summary. At each step, we select a fact with the highest importance score. Whenever a fact is selected, we compare the Simpson [1] similarity score of the fact with all the chosen ones from the same and previous days. If the Simpson overlapping score between the current fact and any chosen one exceeds a threshold α, we ignore the fact and select the next important fact for consideration. The process stops when we reach a pre-defined number of K facts in the final summary.

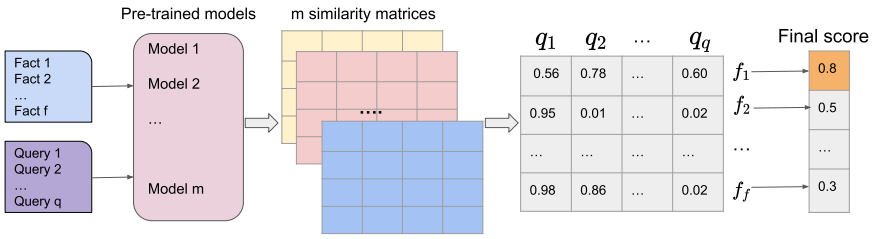


Fig. 1: Overview of steps to compute importance scores of facts.

Graph-based summarization In this summarization approach, we assume that all the facts returned by DFReeKLIM are somewhat relevant to queries of the event day. We focus on finding the most informative and diverse facts based on a semantic graph. Similar to the previous approach, we employ M language models pre-trained on sentence similarity tasks to extract contextualized semantic embeddings of facts.

Next, we construct a graph $G = (V, E, W)$ of facts, where the node set V represents fact texts from previous summaries and all facts of the current event

day. An edge $(v_i, v_j) \in E$ connects two nodes v_i and v_j in the graph. $w_{ij} \in W$ is the average cosine similarity scores between different embedding representations of two facts v_i, v_j. We zero out the weight of an edge if it connects a chosen node v_i from any previous summary and a node v_j, and $w_{ij} > \beta$. Then, a centrality score is computed for every node in the graph.

$$centrality(v_i) = \sum_{j \in \{V \setminus i\}} w_{ij} \tag{1}$$

We consider all facts of the current event day in chronological order. First, a node with the highest centrality score is included in our summary. Whenever a node v_i is selected, we zero out all the edge weights from v_i to any node v_j if $w_{ij} > \beta$ and update the centrality scores. The selection process stops when we reach a pre-defined number of K facts in the final summary.

3 Experiments and Results

3.1 Experimental setup

We run our experiments on the dataset provided by CrisisFACTS track [2]. We consider all content texts from Twitter, Reddit, and Web News. The following models pre-trained on various sentence similarity tasks [11] are used to extract semantic representations of facts and queries.

- **Semantic Text Similarity:** stsb-roberta-large, stsb-distilroberta-base-v2
- **Paraphrase Detection:** paraphrase-xlm-r-multilingual-v1, paraphrase-distilroberta-base-v2
- **Natural Language Inference:** nli-roberta-large
- **Passage Retrieval**: bert-base-nli-max-tokens, msmarco-roberta-base-v2

Thresholds α and β in our query-based and graph-based summarization methods are set to 0.75 and 0.95, respectively. Besides, we select maximum $K = 200$ facts as a final summary for each event day.

3.2 Metrics

We report our results using two sets of metrics introduced by CrisisFACTS track:

- **Metric 1** (Standard metrics): The top-k facts of system summaries for each system's event-day pair are compared against extant summaries using ROUGE or BERTScore metrics. The extant summaries include the summaries extracted from Wikipedia articles, ICS209 reports, or NIST-based summaries. k is defined by CrisisFACTS organizers.
- **Metric 2** (Fact-matching): Given S and F are the set of ranked facts in a system summary, and the gold summary produced by NIST assessors. the comprehensiveness and redundancy ratio metrics are defined as below:

- **Comprehensiveness**: Amount of gain for each unique fact in the gold summary $f \in F$ covered by our system summary.

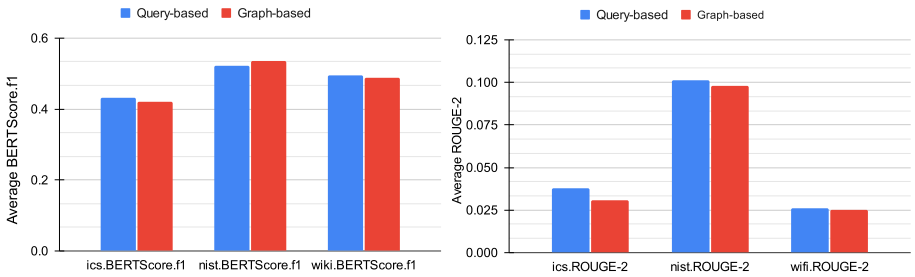
$$C(S) = \frac{1}{sum_{f \in F} R(f)} \sum_{f \in F : M(f,S) \notin \emptyset} R(f) \tag{2}$$

where $M(f, S)$ is the list of system facts that match the fact f. $R(f)$ is the gain assigned to the fact f, which is set to 1 for this year.

- **Redundancy ratio**: number of unique facts that match a gold-standard fact divided by the total number of fact matches present.

$$R(S) = \frac{\sum_{f \in F : M(f,S) \notin \emptyset} R(f)}{\sum_{f \in F} R(f).|M(f,S)|} \tag{3}$$

3.3 Results



(a) *BERTScore* (b) *ROUGE-2*

Fig. 2: Evaluation of our system summaries using standard metrics. *ics*, *nist* and *wiki* denotes summaries extracted from ICS209 reports, NIST-based assessments and Wikipedia articles.

Figure 2 illustrates the average results of our system summaries across all event-day pairs under standard metrics. Our query-based and graph-based summarization models obtain competitive performance in terms of both BERTScore and ROUGE-2, and across all extant summaries. The query-based method tends to show slightly better results compared to the graph-based model.

In Figure 3, we compare performance of our two proposed approaches using fact-matching metrics. It is evident that the graph-based model retrieves a significantly higher number of matched facts compared to the query-based method. However, the set of facts returned by the graph-based method is less comprehensive and more redundant. It indicates that the model fails to cover diverse facts in the gold summaries. This could be influenced by our hyper-parameter setup and the ignorance of query-fact relevance scores in graph building and facts ranking.

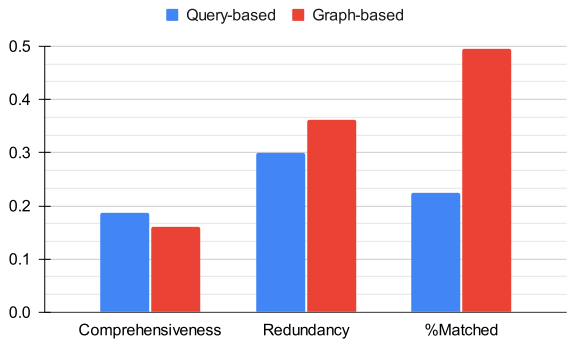


Fig. 3: Evaluation of our system summaries using fact-matching-based metrics.

4 Conclusion

This study introduces two different approaches to generate temporal summaries of crisis events from multi-streams. We apply two-step summarization methods. First, we employ a sparse retrieval method to retrieve relevant facts. Then, we rely on semantic similarity between fact-query or fact pairs computed using different pre-trained models trained on various sentence similarity tasks to extract event summaries. In the future, we will investigate various traits of different streams to fine-tune our summarization results.

Acknowledgements. This work is partially funded by the German Research Foundation (DFG), project Managed Forgetting (NI-1760/1-1), and the European Union's Horizon 2020 research and innovation programme, CRiTERIA project, under grant agreement No. 101021866. Further, this work is supported in part by the Science and Engineering Research Board, Department of Science and Tech- nology, Government of India, under Project SRG/2022/001548. Koustav Rudra is a recipient of the DST-INSPIRE Faculty Fellowship [DST/INSPIRE/04/2021/003055] in the year 2021 under Engineering Sciences.

References

1. Overlap coefficient, https://en.wikipedia.org/wiki/Overlap_coefficient, (Last edited on May 07, 2021)
2. 2022 TREC CrisisFACTS Track (2022), https://crisisfacts.github.io/
3. Akula, R., Garibay, I.: Sentence pair embeddings based evaluation metric for abstractive and extractive summarization. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022 (2022)
4. Amati, G., Amodeo, G., Bianchi, M., Marcone, G., Bordoni, F.U., Gaibisso, C., Gambosi, G., Celi, A., Nicola, C.D., Flammini, M.: Fub, iasi-cnr, UNIVAQ at TREC 2011 microblog track. In: Proceedings of The Twentieth Text REtrieval

Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011. NIST Special Publication (2011)
5. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. 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 2019, Hong Kong, China, November 3-7, 2019 (2019)
6. Macdonald, C., Tonellotto, N., MacAvaney, S., Ounis, I.: Pyterrier: Declarative experimentation in python from BM25 to dense retrieval. In: CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021. pp. 4526–4533 (2021)
7. McCreadie, R., Macdonald, C.: Relevance in microblogs: enhancing tweet retrieval using hyperlinked documents. In: Open research Areas in Information Retrieval, OAIR 2013, Lisbon, Portugal, May 15-17, 2013. pp. 189–196 (2013)
8. Nallapati, R., Zhai, F., Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA (2017)
9. Nguyen, T.H., Rudra, K.: Rationale aware contrastive learning based approach to classify and summarize crisis-related microblogs. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022 (2022)
10. Nguyen, T.H., Rudra, K.: Towards an interpretable approach to classify and summarize crisis events from microblogs. In: WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022. pp. 3641–3650 (2022)
11. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019)
12. Zheng, H., Lapata, M.: Sentence centrality revisited for unsupervised summarization. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers (2019)