Discussion

1. EDA of Dataset

The **CrisisFACTS Dataset** supports research on information needs during crises through two key components: the Event Dataset and the Query Dataset. The Event Dataset contains over 1.2 million crisis-related documents with metadata like event names, timestamps, and source types, enabling analysis of information flow during events.

The Query Dataset includes 922 user-generated queries linked to events, categorized by intent (e.g., fact-finding or assistance requests) with metadata like indicative terms and descriptions. Together, these datasets provide a foundation for studying information retrieval and summarization in crisis scenarios. The following EDA explores their structure and insights.

- Event Dataset overview

  The **Event Dataset** contains a total of 1,220,703 entries with 9 columns. Each entry corresponds to a document associated with a specific event, identified by the `event` and `EventName` columns. The dataset provides detailed information such as the text content (`text`), its source (`source` and `source_type`), a unique document identifier (`docno`), and the timestamp (`unix_timestamp`) when the document was generated. The dataset is comprehensive and complete, with no missing values in any column.

  Event Text Content Overview

Event	Text	Source Type
001	Live updates: San Diego County fire is 92 percent contained	News
001	The Lilac fire now 92 percent contained, Cal Fire officials said Tuesday morning.	News
001	The county of San Diego has opened a Local Assistance Center to help victims of the fire begin the rebuilding and recovery process.	News
001	The center is at the Vista branch library on 700 Eucalyptus Avenue and will be open from 9:00 a.m. to 6:00 p.m. Services offered will include crisis counseling and referral services.	News
001	Homeowners also will be able to get information on residential rebuilding and consumer fraud.	News

The `text` column in the event dataset contains detailed descriptions of updates, reports, and announcements related to crisis events. The entries provide various information about the event, such as live updates, recovery processes, or available services. These texts are extracted from sources like Reddit, Twitter, Facebook and News, and it will be used later in this project for information retrieval and summarization.

- Query Dataset overview

  The **Query Dataset** consists of 922 entries across 13 columns, where each row represents a query linked to an event. Key columns include `query_id`, `text` (the query itself), and `indicative_terms` (keywords representing the query's focus). While most columns are fully populated, the `trecis_category_mapping` column has some missing values, indicating incomplete categorization for certain queries. Additionally, only 409 queries include information about the dataset they reference (`event_dataset`).

  Query Text Content Overview

query_id	text	event_id
CrisisFACTS-General-q001	Have airports closed	CrisisFACTS-001
CrisisFACTS-General-q002	Have railways closed	CrisisFACTS-001
CrisisFACTS-Wildfire-q001	What area has the wildfire burned	CrisisFACTS-003
CrisisFACTS-Hurricane-q001	What is the hurricane category	CrisisFACTS-004
CrisisFACTS-Tornado-q001	What storm warnings are active	CrisisFACTS-017

The query text in the dataset represents user-generated questions or information needs during crisis events. These queries are designed to retrieve relevant information about specific aspects of the crises, such as their impact, status, or recovery efforts. They are categorized into **general queries**, which address broader topics like transportation disruptions or evacuation updates, and **specific queries**, which focus on particular crisis types, such as wildfires, hurricanes, accident, flood and tornadoes. The query text will be used later in the project pipeline to help with our information retrieval.

- Days, Queries and Documents Count for Each Event

Event	Event Name	Unique Days	Query Count	Document Count
CrisisFACTS-001	Lilac Wildfire 2017	9	52	51,015
CrisisFACTS-002	Cranston Wildfire 2018	6	52	30,535
CrisisFACTS-003	Holy Wildfire 2018	7	52	32,489
CrisisFACTS-004	Hurricane Florence 2018	14	51	120,784

The merged chart highlights the variability in the number of unique days, queries, and documents across different crisis events in the CrisisFACTS dataset. Events like **Hurricane Florence 2018** and the **Beirut Explosion 2020** stand out with a high number of documents (120,784 and 186,900, respectively), reflecting their significant information impact and media coverage. Conversely, smaller-scale events such as **Hurricane Laura 2020** and the **2018 Maryland Flood** have fewer documents (18,161 and 13,000, respectively). Most events have a consistent number of queries (around 51–56), indicating a balanced level of information need across crises, while the number of unique days varies significantly, with some events spanning only 2 days (e.g., **Hurricane Laura 2020**) and others lasting up to 14 days (e.g., **Hurricane Florence 2018**).

2. Ranking / Reranking Process

- **Ranking model comparison** Using PyTerrier's pre-stored initial indexes, we conducted a detailed analysis to evaluate and compare various ranking models, focusing on both efficiency and accuracy.

  For efficiency, we measured the memory consumption and processing time required for each model to handle a single data unit (representing one day of a specific event). Accuracy was assessed by using the rankings generated by re-ranking models as the benchmark, under the assumption that these models provide the most precise results. We calculated the Mean Squared Error (MSE) to determine how closely each model's rankings aligned with the re-ranked benchmarks.

  The results, summarized in the accompanying table, reveal that BM25 emerged as the top performer. It had the lowest MSE, indicating the highest accuracy, and also proved to be the fastest model. While its memory usage was slightly above average, its combination of speed and precision made it the most suitable choice for our experiment, leading us to adopt BM25 as the default model.

model	memory	time	MSE
BM25	50.781250	6.514025	216.637500
DFRee	32.414062	11.001296	389.604167
DPH	38.562500	9.705055	404.720833
DirichletLM	85.347656	10.221884	297.254167
Hiemstra_LM	37.957031	9.004418	199.220833
InL2	89.050781	9.054682	222.350000
PL2	42.320312	6.021144	218.820833

- **Reranking model comparison** Since there was no ground truth available for measuring accuracy with reranking, we relied on research papers for model comparisons and focused on those discussed in lectures. We tested the ColBERT model and a mono-T5 model, both available in the PyTerrier package. Referring to the documentation on

Huggingface, we selected a mini ColBERT model as our primary re-ranking approach, balancing efficiency while benchmarking it against the mono-T5 model for comparative analysis.

model	memory	time
COLBERT	798.367188	13.036193
T5	1298.781250	32.529773

3. Summarization Process Having Ranked-Reranked dataset, we experimented with three pre-trained LLMs for the summarization process: Facebook BART, Google Pegasus, and OpenAI GPT. The dataset is groupby request ID, query ID, and event attributes to get a grouped document text. This grouped document text will be passed into the summarization model to get a one single text regarding that specific request ID, date, and query. To choose which model to use for the package, we focused on one event for comparison.

Facebook BART and Google Pegasus are implemented through Hugging Face Transformers. Grouped document texts are cleaned, removing hashtags, URLs, and mentions. OpenAI GPT, however, adopts a slightly different approach. Since we have a `question` column, we asked the model to answer that question using the grouped document text. If the question is answerable, we asked the model to produce the most concise answer as possible. If not, it returns "unanswerable".

- Summarization Process Example

	question	query	bart_summary	pega_summary	gpt_summary
20	What roads are blocked / closed	tree block road closures	Motorists are being warned to expect delays after a number of crashes on the I-805	Accident, three lanes blocked in on 8 EB at Severin Dr, stopped traffic back to	The two right lanes on 8 EB at Severin Dr, the three right lanes on 8 EB at Severin Dr, the two right lanes on I-805 NB after Murray Rdg Rd, and the left lane on 15 SB before 5 are blocked.
2	Have water supplied been contaminated	water supply	Firefighters in California have been using water from a nearby lake to fight wildfires.	A UH-1Y Venom refills its bucket firefighting system with water from Lake ONeill at Camp	unanswerable

4. Evaluation

To evaluate and compare the performance of these summarization models, we followed the methodology outlined in the referenced paper. Specifically, we utilized a Wikipedia summary as a ground truth reference and assessed the model outputs using both BERTScore and ROUGE metrics. Each metric provided precision, recall, and F1 scores, enabling a comprehensive evaluation of the models' performance. As shown in tables below, OpenAI GPT model performed the best for our project due to its speed, minimal memory usage, and ability to produce concise summaries. In terms of evaluation, there was not much of difference across the model. BERTScore uses the contextual embeddings so we got higher scores because the query/questions are related to the grouped document texts. Yet ROUGE measures overlaps using n-grams which will decrease the score since the models may not use the same words as the original text. Ultimately, we decided to use OpenAI GPT for our project.

- Time and Memory Usage Table

Model	memory	time
Facebook BART	1362.03	479s
Google Pegasus	59.53	615s
OpenAI GPT	36.47	187s

- Evaluation Table

Model	Metric	Precision	Recall	F1 Score
Facebook BART	BERTScore	0.72	0.75	0.73
Google Pegasus	BERTScore	0.72	0.74	0.73
OpenAI GPT	BERTScore	0.70	0.73	0.71
Facebook BART	ROUGE	0.006	0.32	0.01
Google Pegasus	ROUGE	0.004	0.29	0.01
OpenAI GPT	ROUGE	0.01	0.15	0.03

5. Challenges / Further improvements Our pipeline, consisting of ranking, re-ranking, and summarization steps, presented challenges when experimenting with every possible combination and hyperparameter to optimize performance. Adjustments ranged from straightforward changes, such as swapping models at each step, to more intricate refinements, including tweaking hyperparameter values to decimal precision or performing prompt engineering for the LLM-powered summarization model. Given the current output often includes summaries labeling questions as “unanswerable,” it is crucial to ensure the model's robustness. This would help build user confidence in the system's responses, particularly when it determines that a query truly cannot be answered.