Fast Extractive Summarization, Abstractive Summarization, and Hybrid Summarization for CrisisFACTS at TREC 2023

Violet Burbank [1], John M. Conroy [2], Sean Lynch [1], Neil P. Molino [2], and Julia S. Yang [1]

[1] Department of Defense
[2] IDA Center for Computing Sciences

February 5, 2024

Abstract

The CrisisFACTS task seeks to find relevant, non-redundant information for an ongoing natural disaster. The task this year allowed both extractive and abstractive summaries. This notebook describes our three submissions: an extractive approach using the `occams` summarizer and two abstractive approaches using GPT-3.5. Of the two abstractive submissions, one used GPT-3.5 on a high-scoring subset of the data, while the second was a hybrid, a paraphrase of an `occams` extractive summary.

1 Introduction

Our focus for the 2023 CrisisFACTS task was to investigate how an existing extractive summarizer algorithm `occams` [3] would help in the CrisisFACTS task. We also investigated using GPT-3.5 [2] to produce fluent text for two of our submissions. Our submissions used a list of relevant documents retrieved by `pyterrier` [1]. As provided in the sample notebook, the `importance_count` score was used to help sort the results or reduce the number when needed. We utilize the GPT-3.5 model for abstractive generation and paraphrasing.

In the following sections, we detail the methods and the three submissions from our group for CrisisFACTS 2023.

2 Methods

Our approach was to take as input the `pyterrier` relevant documents for each `RequestID` and then pass the results to a summarizer, which in our work was

an extractive summarizer, an abstractive summarizer, or a hybrid extractive-abstractive summarizer. The retrieved results are sorted in descending order by `importance_count` with the primary key and `unixTimeStamp`.

1. Our extractive summarizer `occams` approaches the problem of extractive summarization as a weighted bounded maximum coverage problem. This method is very fast, and sentence selection is a linear time approximation to this NP-hard problem. The inputs to the summarizer are documents, a bound for the length of the summary, and term weights. The documents are those returned by `pyterrier`. The summary length was set to be 20000 characters, which was chosen based on 2022 data set ground truth NIST summaries. Finally, `occams` has options for term weights, but for this application we used a fairly robust method of log of the occurrence, where a term is the default for English of stemmed bigrams.

2. For abstractive summaries, the list of sorted documents are trimmed if necessary so their total length does not exceed a prompt window of 15K tokens. The prompt:

   ```
   ``````

   You are an abstractive summarizer that follows the output pattern:

   Text: {text}

   Summary:

   ''''''
   ```

3. The hybrid abstractive summaries are created by prompting GPT-3.5 to paraphrase the `occams` extractive summary. With some experimentation, the following prompt was chosen:

   ```
   ``````

   Please rewrite the following into a coherent and readable paragraph. Do not deviate from the facts of these sentences or add any new information. Follow the output pattern:

   Text: {text}

   Summary:

   ''''''
   ```

As the submissions were required to give a link back to the originating document, this posed a small challenge for the abstractive and hybrid submissions. For this task, we opted for simplicity and used the string matching package `fuzzywuzzy` to find the closest match to each summary sentence in the documents.

# 3 TREC CrisisFACTS 2023 Submissions

## 3.1 A Summary of the IDACCS Three Submissions

Here, we briefly analyze the results of our three submissions to CrisisFACTS 2023. The three submissions were submitted with the following labels and descriptions.

1. **Run identifier:** IDACCS_occams_extract

   **Run type:** Automatic [listed as Manual]

   **Uses TREC-IS category labels?:** no

   **Calculating importance:** We use an extractive summarizer `occams` to select the most representative sentences from the `pyterrier` run. The importance score for the ith sentence is $s_i = (n - i)/n$

   **Data streams:** Twitter Reddit Web news

   **Extractive or abstractive?** extractive

   **Description of run:** `occams` is an extractive summarization system that approximately solves the bounded maximal coverage problem. We used bigrams with the LOG_COUNTS term weighting scheme.

2. **Run identifier:** IDACCS_occamsHybridGPT3.5

   **Run type:** Automatic [listed as Manual]

   **Uses TREC-IS category labels?:** no

   **Calculating importance:** The importance score for the i-th sentence is $s_i = (n - i)/n$ in the summary.

   **Data streams:** Twitter Reddit Web news

   **Extractive or abstractive?** abstractive

   **Description of run:** We use a hybrid approach that generates an extractive summary via `occams` and then uses GPT-3.5 to generate a summary, a paraphrase of the `occams` extract.

3. **Run identifier:** IDACCS_GPT3.5

   **Run type:** Automatic [listed as Manual]

   **Uses TREC-IS category labels?:** no

   **Calculating importance:** The importance score for the i-th sentence is $s_i = (n - i)/n$ in the summary.

   **Data streams:** Twitter Reddit Web news

   **Extractive or abstractive?** abstractive

   **Description of run:** We used GPT-3.5 to generate a summary, then segmented and found the best matching `factText` for attribution.

### 3.2 Comparison of the Three Submissions

We give a brief comparison of the three submissions, IDACCS_occams_extract, IDACCS_occamsHybridGPT3.5, and IDACCS_GPT3.5, which for this discussion we use the short labels `occams`, hybrid, and GPT. We focus on the NIST summaries for the automatic metrics and the manual evaluation of these submissions based on the NIST summaries.

In Figures 1 and 2 we give the scatter plot of ROUGE precision and recall for the submissions. The scatter plot gives the ten events (CrisisFACTS-009 through CrisisFACTS-018) for the three submissions. The plot shows that the hybrid run outperformed the GPT run in precision and recall. As the `occams` extracts were the longest of the three submissions, it is unsurprising that it dominated recall and lagged in precision.

Figures 3 and 4 give the corresponding scatter plots of the three submissions for BERTScore precision and recall metrics. As with ROUGE scores, the hybrid submission dominates the GPT submission in BERTScore precision and recall, but noteworthy is that hybrid dominates `occams` even in BERTScore recall. We further note that when testing for statistical significance with a Wilcoxon paired testing the *p*values are both less than 0.01, so these differences in precision and recall are not likely due to chance. BERTScore favors the more concise abstractive summaries of the hybrid over the `occams` extracts of which the hybrid is paraphrased.

Finally, we give the scatter plots for manual assessment in Figures 5 and 6. The results are for the 58 `RequestIDs`. (Each of the ten events averaged 5.8 `RequestIDs`). The results are similar to BERTScore in that the hybrid approach is generally favored in precision and recall. Note that human annotators evaluated just the `occams` extracts and hybrid submissions. We include a third series, the mean for priority 1 and 2 for all participants. The plot suggests that the hybrid approach is significantly below the mean manual assessment for redundancy and significantly better in redundancy removal. Indeed, when checking with a Wilcoxon pair test both these differences are significant with a $p$ value of 0.01 or less, so they are not likely due to chance.

## 4 Conclusions

The first objective of our submissions was to explore the effectiveness of a bounded maximal coverage extract summarizer, `occams` for the CrisisFACTS extractive task. `occams` generally outperformed GPT-3.5 in the automatic metrics of ROUGE and BERTScore. It is noteworthy that it outperformed GPT-3.5 in both precision and recall, despite `occams` summaries being longer.

Our secondary objective was experimenting with a hybrid approach, using `occams` in conjunction with GPT-3.5 to achieve a lower-cost hybrid summarization method. The hybrid (`occams`/GPT-3.5) summaries outperformed the `occams` extracts. We hypothesize that the paraphrasing of `occams` extracts removed redundancy by combining multiple facts discovered by `occams`.

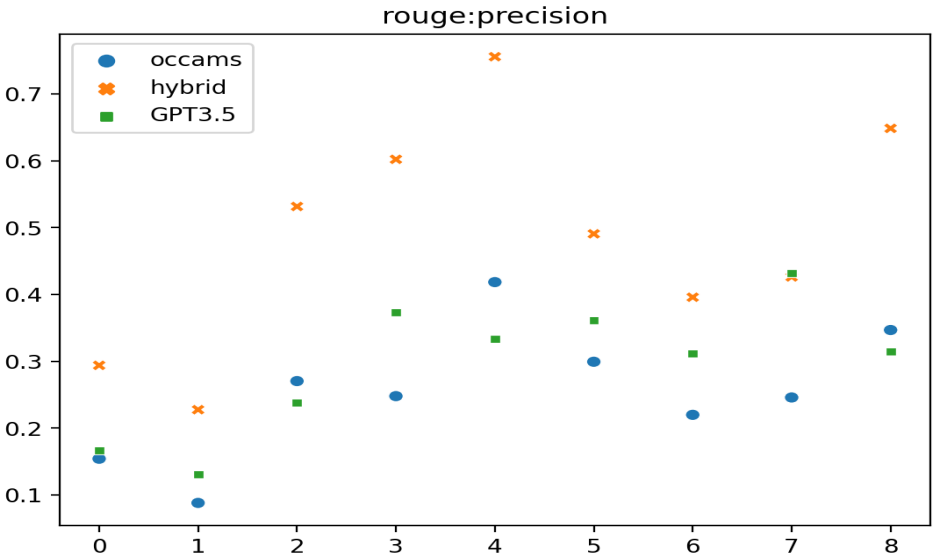


Figure 1: ROUGE Precision scores for `occams`, hybrid, and GPT

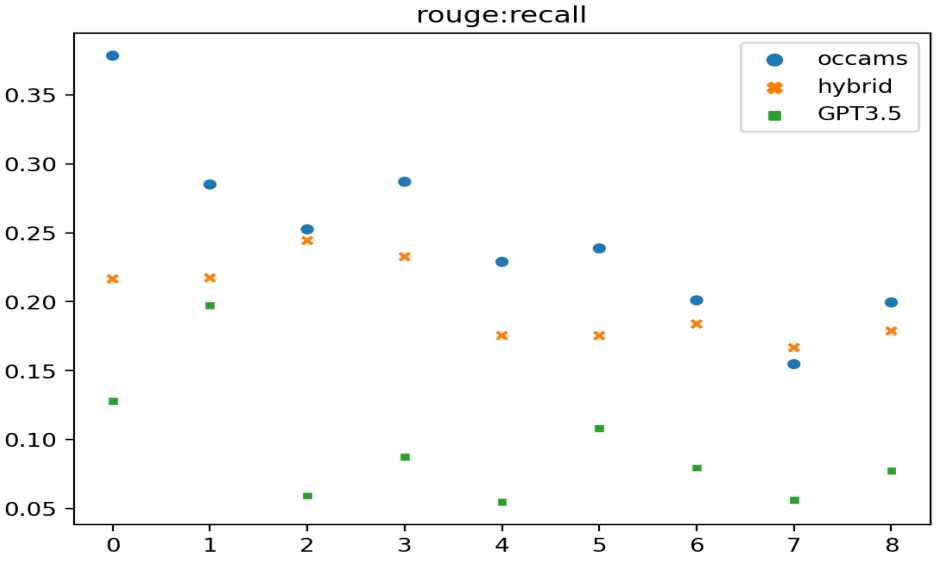


Figure 2: ROUGE Recall scores for `occams`, hybrid, and GPT

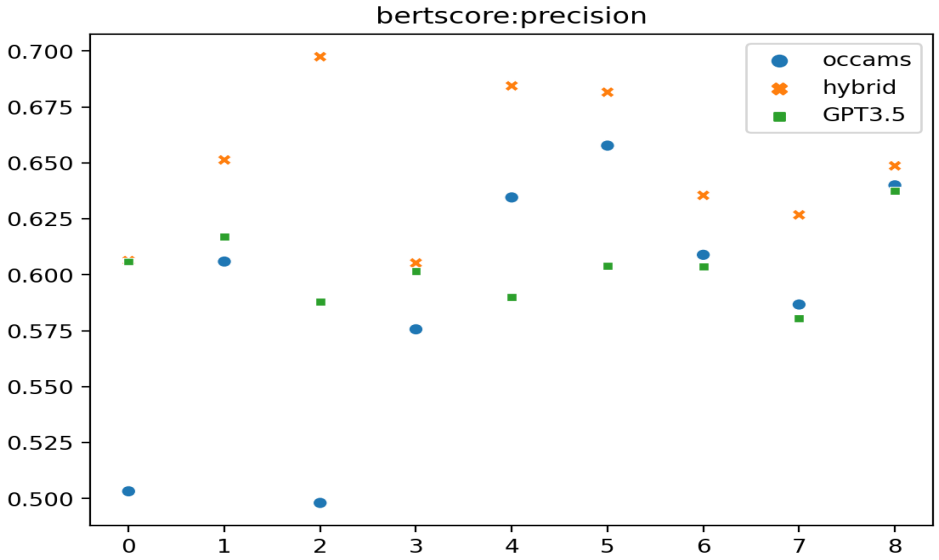


Figure 3: BERTScore Precision scores for `occams`, hybrid, and GPT

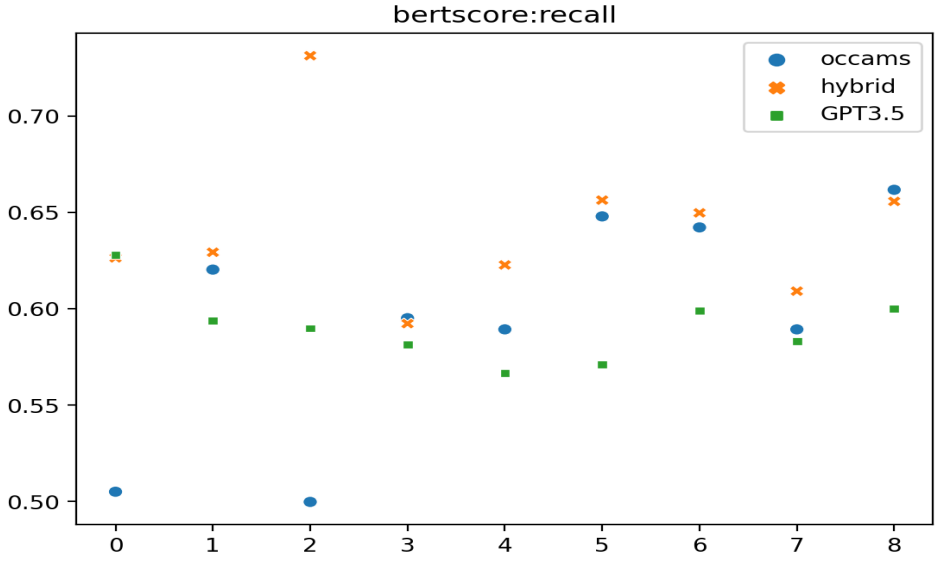


Figure 4: BERTScore Recall scores for `occams`, hybrid, and GPT

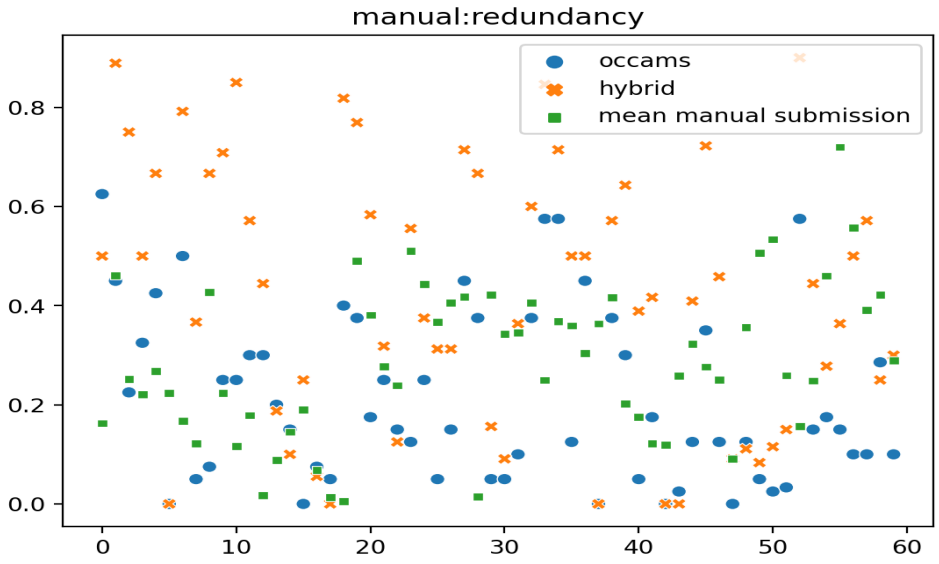


Figure 5: Manual Redundancy (Precision) scores for `occams`, hybrid, and GPT

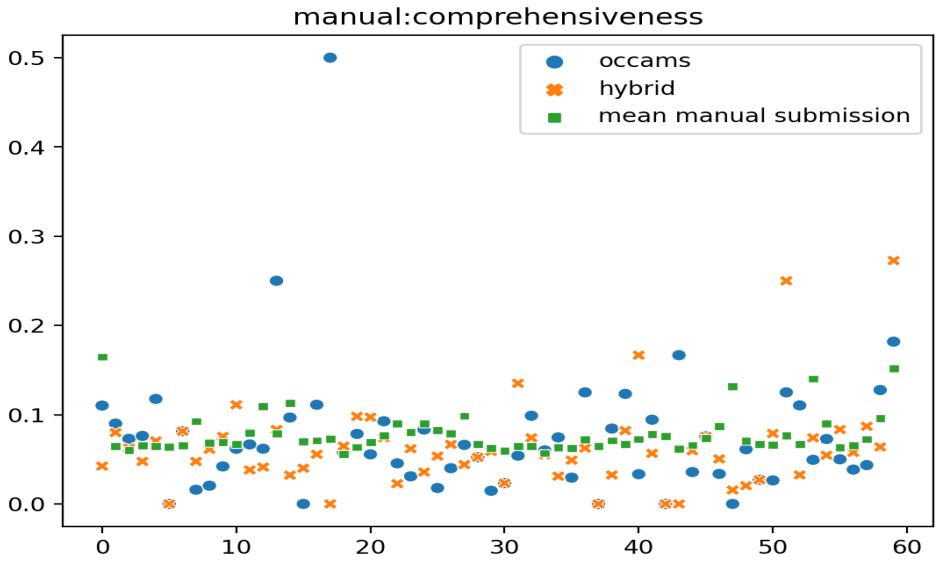


Figure 6: Manual Comprehensiveness (Recall) scores for `occams`, hybrid, and GPT

We find the results encouraging that combining extractive summarization with paraphrasing outperformed abstractive summarization on the CrisisFACTS track data. As a bonus, since extractive summarization with `occams` is linear in the length of the text, the hybrid approach is much cheaper than using GPT-3.5 alone. The hybrid summaries took roughly 1/10 the time to compute than the GPT-3.5 summaries.

# References


[1] Craig Macdonald and Nicola Tonellotto. Declarative experimentation in information retrieval using PyTerrier. In *Proceedings of ICTIR 2020*, 2020.

[2] OpenAI. GPT-3.5 language model (version `gpt-3.5-turbo-16k-0613`). `https://openai.com`.

[3] Clinton T. White, Neil P. Molino, Julia S. Yang, and John M. Conroy. `occams`: A text summarization package. *Analytics*, 2(3):546–559, 2023.