Midterm Report

Massey University 159333 - TAIC report summarizer

James Thompson (21011195) - 1jamesthompson1@gmail.com

11/09/2023

Introduction

Work and progress has been steady over the first half of the semester. The project is being developed openly here. In the proposal I discussed the three deliverables. The first one has been completed and the second one is in progress due be completed on the 25th. The third deliverable is set to be defined after feedback from the second deliverable is received.

Completed tasks

There have been three main tasks completed so far. Framework, basic summarizer and basic theme generation. Ontop of this work has been commenced on the viewer app discussed viewer app.

Framework

By framework I am mean the skeleton of code that is there to allow the summarizer and theme generator to plug into. This "skeleton" also includes the support for downloading, parsing and then looping through all of the reports. But in general the work here can be seen in the Gather_Wrangle module. This module is the simplest of the three modules and is the most complete. Various other scripts/classes have been made to help speak with the openAI api and managing the various input/output files and folders; Config, openAICaller, Themes and OutputFolderReader.

Basic summarizer

I have made a module which purpose is read and summarize the reports. It has 3 distinct stages 1. Using regex to find the content section of the report 2. Using the openAI api to read the content section and find the pages for the Analysis and Findings sections. 3. Using the openAI api to read the Analysis and Findings sections and then provide summary.

The summary of a report is currently only weightings of predefined themes, with the summary expected to deepen in the third deliverable. This means the model is provided with a list of theme titles and descriptions and then will return a percentage for each theme proportional to how important the theme was to the accident.

Basic theme generation

After initially hand writing the themes (#3) work was done to automate this process (#14). This resulted in another module added. Adding in the theme generation is important but also challenging because it adds an entire other dimension to the output of the engine. Meaning that the output is not only a list of weighting for themes but the themes themselves.

Theme generation works by using openAI api to create a summary of the themes present in each report. These summaries for each report are then put together and the openAI api is asked to find 5-10 themes that are present in all of the reports. These themes are then used as the themes by the summary step to assign weightings to.

Remaining tasks / Challenges

There are broadly three remaining tasks.

Convergence / Variability

Firstly is arguably the most important. Reliability of the engine and guaranteeing the output is correct is outside the scope of the project. However making sure that the output is consistent and converges to some values will be quite important. Due to its importance this is the main goal of the second deliverable.

The inconsistency is present because of the inherent randomness of a gpt 3.5 response. This means that various methods will need to be employed to make the responses which are either in natural language or just number semantically constant.

Summarization

As the goal of getting weights for each report is to do after the fact statistic analysis on the generated csv. For that analysis to have any meaning it needs to not differ each time. Fortunately as the output is numeric we can use multiple responses and take the mean of them. This helped with some variability but not enough.

Moving forward I first hope that access to gpt 4 will provide more consistent answers, along with the use of the temperature parameter. If this is not enough then I will need to look into other methods to decrease the variability and create the effect of convergence.

Theme generation

The summarization and weight assigning is fully dependent on the theme generation. The theme generation is currently not converging. Each run through of the themes generates difference themes both in quantity and quality. Unlike the weightings numerical techniques cannot be used to get the "mean" response. Instead I will need to look at fine tuning the model parameters as well as moving from a one-shot approach to a few-shot approach. This will allow the model to learn from the themes that have already been generated and hopefully converge to a set of themes.

viewer app

As discussed here the viewer app is a deliverable that has been added to the project. This will be a Shiny app that will allow the user to view the output of the engine in a user friendly way. This will be a simple app that will allow the user to select a report and then view the summary and themes. It is currently at the point of having a basic search function. Work will be done before the second deliverable so that the viewer is more complete, i.e making the search fuzzier (#23)and the output table complete (#22) and interactive (#19).

It is being made use Shiny for python as it is the simplest to get a basic web app up and running. I am using a web app so that hte barrier for usage is as low as possible (simply click on the given url).

Extraction of more information

As discussed in the initial proposal there will need to be the addition of more information from each of the reports. This will allow the viewer app to provide more information to the user. I expect this work to be completed in the third deliverable. In practice this will likely include the addition of more variables to the output csv, for example fatalities, location etc.

Scope and and Objective changes

The scope of the project remains mostly unchanged. The two changes that have been made are the addition of a Shiny app and the second and third deliverable goals.

Shiny app

In its current state there is no way for a non-technical person to easily appreciate what the engine is actually capable of and the utility it provides. The best way to fix this is by creating an example app that can be used to demonstrate an application of the engine and what it can do. Thus the goal of this app is to make the output of the engine accessible. By allowing you to search and filter the results plus visualize the generated themes and weightings. This means that along with the engine widening the third deliverable will involve development of the viewer app.

Engine widening

Each deliverable has had a slight change in what it entails. I now expect that the goals mentioned above regarding convergence/variability and viewer app will take precedent and by the focus of the second deliverable. While the third deliverable will focus on widening the output csv and the viewer app to include more information from the reports.

In theory this process should be relatively straight forward as it will simply involve asking the model to generate more simple answers to question such as “how many fatalities” or “What category of location was the accident in”. Then adding these extra features to the viewer app should not be a problem.

Yet rather than define it formally I will let the feedback from investigators using the viewer app to guide what more information they would like.

Conclusion

As can be seen progress has been made. The engine is in a state where it can be used to generate summaries and themes. Currently consistency and accessibility are the two pain points making them the focus for the 25th of September deliverable. While the final step will be to make it more useful through more variables and a more fleshed out viewer app.