**AI-Driven**

**News Summary and Verification**

**Software**

**(AKA Oryx Dashboard)**

**CS39440 Project Outline Document**

**G4RR Computer Science and AI**

**Declan R A Wadsworth | 200061043 | drw8**

**Version 0.1 (Draft) | 27/01/2024**

**Supervisor Chuan Lu**

**Project Description**

**Outline**

This project; given a brief outline by the document title page, aims to produce a native piece of software on Windows Desktop (10/11) or possibly as an android app for easier use throughout the day for those not based at a desk. The software will produce a front-end UI that provides the user with news procured throughout the day (based on news interests given by the user), whilst also leveraging emerging AI chatbots such as Google Gemini or ChatGPT, to produce representative headlines to these articles as well as summaries of the articles. Misinformation and misleading/clickbait headlines are one of the biggest problems the world faces, creating divisive social trends and evoking emotional responses from people in order to manipulate readers into believing subjective and/or non-factual accounts. The aim of this software is to create something as trustworthy and as transparent as possible to combat these kinds of social threats (such as what currently occurs in the USA, Russia, etc.).

The UI will break into different containers for the different interests of the user, to produce a logical and easily readable format to consume. It can also include a header providing general information such as local time, weather and any other general pieces of information which can be included if time permits.

**Main Aspects/Features**

The project aims to produce a piece of software that provides multiple useful and relevant features in todays world. These features include, but might not be limited to:

* **Interval Web-Scraping based on user Interests:** Using one of the advanced AI chatbots mentioned (accessed through their API within the software back-end), a user will provide several different news interests (such as technology, British Politics, American Politics, etc.), with the chatbot providing a list of news organizations (most likely their web domains) that provide such articles. These domains can be saved locally either in a database (SQL, etc.) or in a simple text-file with a delimiter to separate each domain. A web-scraping library will find the most recent articles from these domains, and scrape the articles body for use in the software. This entire process occurs repeatedly throughout the day (in-between app closure, the app could either save all current summaries and the last time it updated, or to ignore interval if the app was closed and start the scraping process again) with a user provided interval (possibly selecting from a list of 30 minutes, 1 hour, 2 hours, etc.), to keep a user updated on events throughout the day.
* **Chatbot Summarization and generation of representative headline:** Once the scraping process has occurred, a chatbot (either of those mentioned in the outline section), will read each body provided, and summarize the article (possibly with a maximum character limit to reduce API token usage, but does require further investigation). As well as this summary, the chatbot will tackle a major problem in the world today; that being providing a representative and truthful headline free from any semantic/emotional language and one that is based on the facts/content provided in the article.
* **Traffic-Light System for domain factual integrity and bias:** Several different verifiable sources provide information on the relative bias of news and reporting organizations. Some form of qualitative algorithm will be implemented using data from these kinds of sources, to rate each domain provided by the chatbot (based on user provided interests, see first feature), as being:

**|Green:** No bias, factual with verifiable sources

**|Amber:** Some bias, generally trustworthy but can feature subjectivity

|**Red:** Highly biased, not objective

**Proposed Tasks**

Over the course of project planning, design and development, several different tasks and investigations will need to take place in order to facilitate a professional, relevant and efficient program. These tasks include, but may not be limited to:

* **Investigation/Research of producing trustworthy, reliable and user-relevant program:** This first task will explore the news industry as a whole, looking into what makes an organization trustworthy (aspects such as sourcing, use of semantic vs objective language, etc.), how trustworthiness can be objectively judged as well as how to engage these aspects with a wide range of users having a wide range of views and objective assessment ability. This investigation will give weight into how the software can be designed as well as the portion of the population it will be relevant to. It will also provide insight into producing the proposed traffic-light system, investigating sources/organizations that produce analysis and evaluation of different news providers (some possible sources include MBFC, AdFontes and AllSides).
* **Investigation concerning evaluating different chatbots:** The chatbot is integral to the proposed software, and its ability to provide accurate summaries as well as headlines is crucial to producing a relevant and trustworthy piece of software. There will also be considerations concerning API usage (requests and tokens used over a period to remain in a free-tier). Articles can come in a variety of different lengths and conciseness, so ensuring the software is able to process all articles fetched from the web scraper is crucial to producing something effective to a proposed user. This weighs into API usage, as longer documents can use more free API resources. Limiting the amount of article text provided to the chatbot is a possible consideration, but this could affect the accuracy of summaries and headlines produced.
* **Decision concerning building to Android and/or Windows (Including development language and UI framework):** Seen as the project proposes keeping a user updated throughout the day, considerations concerning target devices must be undertaken. The program needs to be easily accessible to as wide an audience as possible. People who are not at desks throughout the day would require a mobile version, but some may find a desktop version more relevant. Deciding to build to either or both is crucial, as the software must intend to reach as wide an audience as possible; the issues this project targets being a moderate threat to a healthy society. Some UI frameworks and languages allowing building to both, whereas others are not cross-platform or do not clearly separate the front-end UI from backend logic (if deploying to multiple target devices).
* **Planning:** A task that will involve detailed planning for possible implementations of the proposition, including all relevant diagrams and content such as UMLs, flowcharts, etc.
* **Development:** This can be split into the development of the UI and the back-end logic. The back-end logic can be further divided into implementing the different proposed features and connecting these to a chatbot service.
* **Project Meetings and Diary:** Each week, a meeting is conducted with the project supervisor to provide updates on progress and discuss any issues that have presented themselves as well as possible solutions to those. A diary will be kept of what was discussed at these meetings in the form of either a markdown language document or html document.
* **Demonstrations:** Two demonstrations are required for this project. Ensuring the project is at relevant stages in its development for these will be a consideration throughout the project lifecycle. The mid-term demonstration aims to demonstrate the capabilities of the main features discussed, focusing on back-end logic rather than any implementation for a UI. The final demo will include the front-end and back-end working in tandem, representing a deployable piece of software.

**Project Deliverables**

The project requires multiple pieces of work to be produced, including different aspects of documentation, as well as the software itself. These include and are limited to:

* **Deployable Application and Source Code:** The application itself having been built into an executable for the target device/platform as well as an accessible git repository to the application source code.
* **Final Report:** A document that details the full project lifecycle, including all third-party libraries, frameworks and tools utilized.
* **Final Demonstration:** A demonstration meetingto showcase the applications full capabilities. It must be considered throughout development as an end-point.
* **Mid-Term Demonstration:** Another demonstration occurring half-way through the project lifecycle, focusing on back-end logic and implementation. Another consideration when beginning and pursuing development.

**Annotated Bibliography**

**To Be Implemented in a later document version**