IntelliNews: Removing Bias From Modern News

A Comprehensive Solution for Unbiased News Consumption

Prepared By: Eleanor Haas, Yunis Hussein, and Jack Ogden

Virginia Tech, CS 5934, Dr.Mengistu

06/02/24

Purpose of the document

This Inception Report defines the scope of the solution, in compliance with the agreed upon project scope. Our group will use this report to ensure that we all have the same understanding of the initial business requirements. Moreover, this document will also serve as a binding tool between our group and any other outside stakeholders to clarify the scope and units addressed.

Inception Report

Project Name: IntelliNews	
Date: 06/02/24	Revision Number: 1
Date: 06/11/24	Revision Number: 2
Date: 06/17/24	Revision Number: 3
Date: 08/01/24	Revision Number: 4

Version History

Date	Document Version	Document Revision History	Document Author/Reviser
06/02/24	1.0	Initial Draft	Eleanor Haas, Yunis Hussein, and Jack Ogden
06/11/24	2.0	Reviewed and revisited after first stand up	Eleanor Haas, Yunis Hussein, and Jack Ogden
06/17/24	3.0	Added UML diagram to represent the architecture	Eleanor Haas, Yunis Hussein, and Jack Ogden
08/01/24	4.0	Added the remainder technical aspects	Eleanor Haas, Yunis Hussein, and Jack Ogden

Approvals

Date	Document Version	Approver Name and Title	Approver Signature
06/11/24	2.0	Jack Ogden, Project Manager	JO
06/17/24	3.0	Jack Ogden, Project Manager	JO

08/01/24	4.0	Jack Ogden, Project Manager	JO

Table of Contents

Background

In the modern day news has become a challenge to consume. To be an informed citizen it is important to have a good grasp of current events from a neutral, reliable source, and to form one's own opinions on the matter. It is hard to tell what sources are reliable and it is even harder to tell what bias a source might have. People get trapped in “echo chambers,” where they see news with the same spin and cannot break out of it. This leads them to not consider any opinions outside of those biases and increases radicalism.

The goal of Intellinews is to help strip the bias from news by taking news stories from multiple sources and finding the commonalities. The idea is that in each news story there are indisputable facts independent from any bias in reporting. By taking numerous stories on the same topic from different sources with different biases, we can extract those facts and present them to the user without any bias. That is the goal of the app. It will scrape news stories from the web, then condense them down into facts only. If the user wishes to read more, it will provide links to the sources it used.

The customer from this project is the average English-speaking, likely American internet user who wants an easy, simple website to get news from. This is someone who cares about getting unbiased news and not getting stuck on one side of the political spectrum.

Project Objective

The project's objective is to study, design, customize, test, and deploy a tool that will serve as a place for people to read unbiased news. There should be a web scrapping component to retrieve various pieces of qualitative news data. Additionally, the project

should consist of an LLM element to take in the data and manipulate as intended. Lastly, there should be a web application to display the results neatly and concisely.

Technical Aspects Defined

Web Scraper:

We will be using Python, more specifically the BeautifulSoup library, to scrape data from the web. We plan to separate scraped data via genre. Each genre will have its own text file and all the text files will be put into one central repository.

The trending headline web scraping aspect of our code will use the pytrends library (https://github.com/GeneralMills/pytrends?tab=readme-ov-file#realtime-search-trends) to search Google Trends for the top trending stories in either real time or over the past 24 hours.

To search the internet for articles relating to each headline we will use the gnews library (https://github.com/ranahaani/GNews) to search Google News for articles related to a given headline, with more relevant articles appearing at the top of the list.

To scrape the body of the articles we will utilize the BeautifulSoup library in Python. We will also be using the html5lib Python package as our HTML parser.

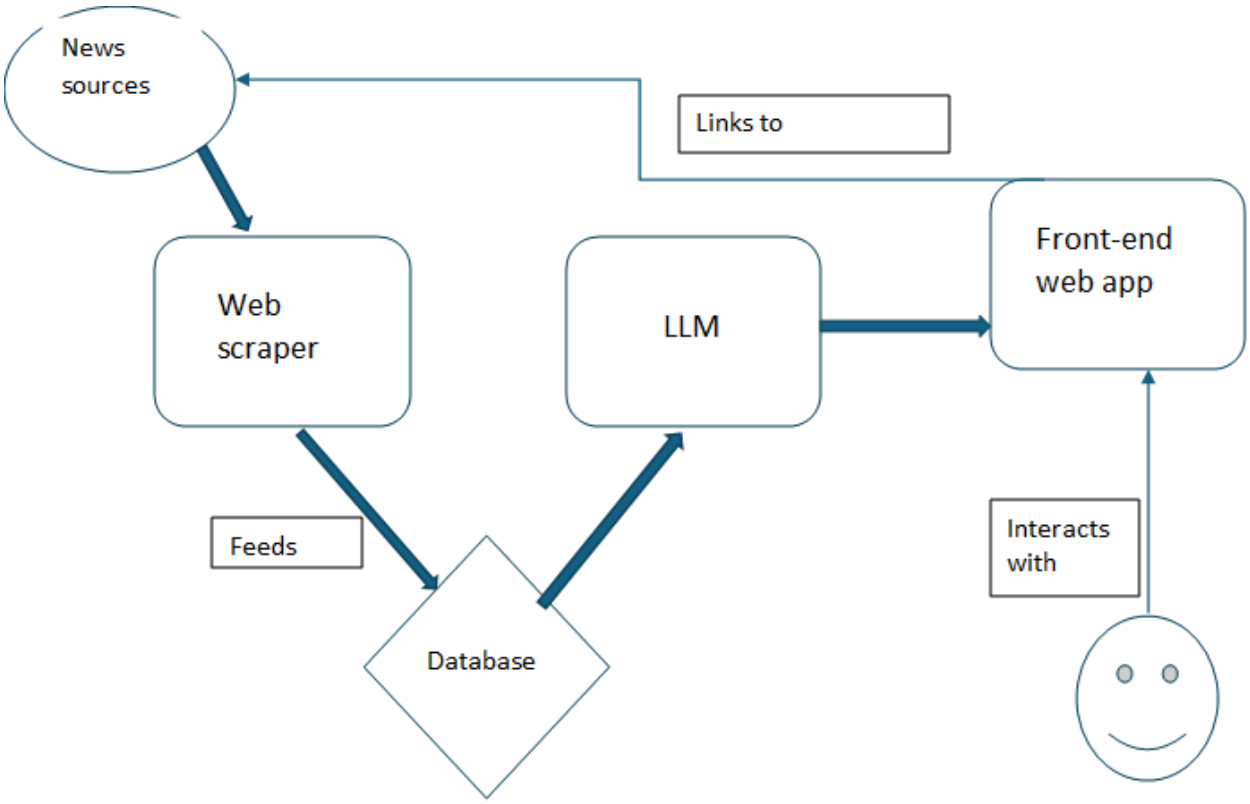
Summarizer:

We will be utilizing various python libraries and summarizing techniques to make this possible. We will output a CSV as our end product for this portion.

Front End:

We will be utilizing the React framework, HTML, and CSS in order to successfully implement a function website. As far as data storage goes, we will simply be using a CSV and reading, storing, and updating aspects from the file. The main languages that we will be using for thisa part is Typescript and Javascript

System Architecture



Project Scope

This project's scope is limited to current events and not any other forms of news or journalism. It will only use news sources that are available online and not behind a paywall. This then excludes any newspapers that are printed only. It also excludes any social media, gossip, or otherwise unofficial news sources that are not guided by journalism ethics. We will also be limiting the project to news sources that are available in English.

The plan for this project also does not include its deployment on a wide scale. It will mainly focus on the development of the app itself and how the users interact with it. It will not focus on how to deploy it for widespread use.

Project Deliverables (Expected Outputs)

Inception Report:

This document outlines the plan we have to complete the project and the methodology we intend to use to do so.

Functional Requirement Specification:
In the functional requirement specifications, we will lay out all the intended features of our project and how we will achieve them. This will then be used to create our task backlog for the scrum methodology.

Sprint Journal Records:
This will be documentation of the work every member of our group completes on a weekly basis from now until the project is finished.

Software package with source code:
The software package will include the web scraper, natural language processor, and the front-end user interface. This will be the meat of the deliverables, which prospective users will have access to the software package as a web application.

User manual:
Since we plan to host our application on the web, only minimal documentation for the user will be necessary. We plan to have a user manual detailing how to navigate and use the site, how to make and delete an account, and how to specify news preferences.

Project closure report:
This report will be a summary of our development process throughout the project, the challenges we faced, and the deviations from the plan we outline in this document.

Project Assumptions and Constraints

Assumptions

When agreeing to take on this project, some assumptions have been made that impact the final product's effectiveness. First, we assume that there is an adequate amount of free news sources that will give us data that our tool can intake. Lastly, we assume that we can freely web scrape the articles necessary without having to go through any prior measures. While these may seem like quite heavy assumptions, it is important to note that we have full proof work arounds in the scenario that these assumptions turn out to be false.

Constraints

The product has a couple of constraints that we hope to address as we move further along in the project. Firstly, data quality is something that can be variable. While the data we have access to is high quality, many of the articles that could be retained from paid news sources tend to have much higher bias. To address this constraint, we may investigate purchasing a couple of subscriptions depending on what our analysis tells us. Secondly, arguably one of the biggest constraints is time. While the ceiling for this project is extremely high, we need to be aware of the short 10-12 weeks we have to complete the entire thing. Lastly, technical limitations are a constraint we need to actively think about. Overall, we can see how important it is to actively think about the available time, tech stack, knowledge, and data we have when working diligently to complete this project.

Team Organization

This team will consist of three members. These are the roles:

- Jack Ogden: Project Manager
- Eleanor Haas: Developer
- Yunis Hussein: Analyst

Every individual in the team will work on every aspect of the project in some capacity, but there will be an individual who will lead certain parts and play more of a supporting role in others. This is the specific breakdown:

Project Manager: Leads and keeps track of overall timeline, deliverables, sprint meetings, and any other high level managerial activities.

Developer: Leads coding, testing, deployment of the final product, and any other technical aspects of the product.

Analyst: Leads information gathering, user interface design, conversation with stakeholders, sprint summaries, and any other tasks that involves stakeholder input.

Project Management Approach

The project managers understand the necessity of applying solid and industry standard project management model. In this regard, we will be using the Agile method of sprints to complete our code and remain flexible to change. We will also use PMI's model which is organized into five groups:

- Initiating: it authorizes the project or a specific phase. This will be done via the project definition document, which outlines the goals, process, and roles for the group.
- Planning: it defines and redefines objectives and selects the best of the different alternative courses of action to attain the objectives that the project or phase was undertaken to address. This will also be done in the project definition and at the beginning of each sprint, so the team members know how to proceed.
- Executing: it coordinates people and other resources to carry out the plan. This will be done over the 12 weeks (about 3 months) of the course by the team members.
- Controlling: it ensures that the project or phase objectives are met by monitoring and measuring progress regularly to identify variances from plan so that corrective measures can be taken when necessary. The team members will do this with weekly meetings and daily communication to ensure that everyone is on the same page.
- Closing: it formalizes the acceptance of the project or phase and brings it to an orderly end. This will be done in the final presentation of the project at the end of the 12 weeks.

Methodology

For this project we will adopt the scrum methodology to guide our development. The development process will begin with creating a backlog of development tasks for the project, which will then be arranged into sprints. We plan to have 3 sprints in the first stage of developing a web scraper, the first sprint will entail scraping the web to find recent major events from recent news headlines, next we will focus on collecting as many articles as possible for a given event, and for the final sprint of the stage we will scrape the text from each article ready for use by our natural language processor. For our next phase we will develop a natural language processor to distill the information from the articles collected by the web scraper, and this stage will be done in one sprint. The final stage of our project, which will also fall under one sprint, is the front end and user interface, which will combine the previous stages and give users a pleasant

dashboard to read the news. Each stage and sprint are designed to have a functioning product as soon as possible, allowing for ample time for fine tuning and iteration.

Risks Management

We would like to incorporate tailored news feeds for users, but this poses potential national security risks if user data is exposed. In 2018 Strava, an exercise tracking app, data was used to locate United States military bases. If our user data was exposed, there is the potential for bad actors to identify high-ranking officials in certain fields based on their news interests. To mitigate this risk, we need to make sure our data is secure and that we have minimal connection between the actual identity of the account holder and the subsets of news they are interested in.

In each phase of our project, we face technical challenges that pose risks to the project completion. In the first stage of our project we will be developing our web scraper, during this phase hurdles will include handling articles behind paywalls, accurate aggregation of headlines, and the accuracy of text scraping on articles. To mitigate the risk of this stage taking too much time we will not hold ourselves to a standard of perfection, by first creating a web scraper that ignores articles with paywalls and does not have perfect text scraping capabilities. Once we have a web scraper which works to some extent, we will take advantage of the agile methodology to further improve the capabilities of the web scraper as time permits.

The second stage of our project is condensing the information in the collection of articles into a concise summary. In this stage we would like to implement a natural language processor which is tuned to our needs of summarizing an event from multiple accounts. In the case that this stage prevents us from moving forward we will adapt our strategy to use an existing large language model to handle the summarization of events. The goal of the final product is to give a non-biased summary of an event to the user. To achieve this, we must make sure that for each event we are capturing accounts from across the political spectrum, and ensure no article is referenced multiple times. To mitigate these risks, we should compare the similarity of articles and discard articles that are incredibly similar.

Detailed work plan

Major milestones

	Milestone	Major Deliverable
	Project definition (by week 2)	Finish project definition assignment
	Complete web scraper (by week 4)	Ensure web scraper can seamlessly scrap articles from the web
	Midpoint presentation (by week 5)	Create slides for presentation Create sample use case to present
	Complete condensation method (by week 7)	Ensure LLM is fully implemented and functional
	Complete front-end (by week 9)	Make sure user interface is fully functional
	Final presentation (by week 12)	Create slides for presentation Application should be integrated fully and functional

Project Schedule

Task	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Project Definition											
Sprint 1/2: Web Scraper											
Midpoint Presentation											
Sprint 3/4: Condensation Method											
Sprint 5/6: Front End Development											
Sprint 7: Integration Between All Parts											
Sprint 8: Final Testing and Evaluation											
Final Presentation											