CS 510 Project Proposal (Development Track)

Course Project Team 9：

- Shuning Zhang (sz31@illinois.edu)
- Rachanon Petchoo (petchoo2@illinois.edu) - Coordinator
- Yushi Li (yushili2@illinois.edu)
- Mayank Umesh Shetty (mshetty4@illinois.edu)

Functions and Users

We plan to implement a news recommendation system as a standalone, web-based application. The major function for this application that we envisioned is that we want to pull fresh news from various APIs and probably a web crawler and present them to the user in order of their preferences. The target users for this application are anyone who craves news that is related to their interests—whether in technology, business, finance, politics, or niche domains—without being overwhelmed by irrelevant content or ads.

Significance

Our proposed system is built to solve the common pain that users experience when they need to browse between news platforms to read about the news or spend time trying to find the article they are interested in. We are aiming to provide a solution that provides timely news with personalized recommendations based on the user's recent activity and interest. Each time the user enters the app, we will run a new scan for the latest news. Also, unlike other recommendation logic, we always want to keep the user's interest in the topic in mind, with a focus on the user's recent hitting numbers, but still, with the recommendation, keep an overall news recommendation. Additionally, we want to create a random push mechanism that shows the current trending topics, ensuring users stay informed on global events or trends even if they may be beyond their usual preference.

Approach

As for our approach, we want to utilize the News API (https://www.thenewsapi.com/) to fetch fresh news articles. We want to convert the news articles into vector embeddings using technologies like BERT and then store them in a PostgresSQL database alongside their metadata (such as authors, user interactions, etc.). As for the news recommendation/personalization part, we want to build user profiles by giving more weight to the news articles that users spent more time on. An example would be to build a user profile by using a direct “weighted average” of the embeddings of the news articles, with the weight being the amount of time that the user spent reading each article. Another approach could be to list out a large number of topics and represent a user profile with a vector that shows the amount of interest that the users have for each topic. With these approaches, we can steer the user profiles toward the news and topics that they are interested in. Additionally, by keeping track of user interactions with each news article, we can identify the “trending” news/topics and randomly suggest it to users as well, so that they can stay informed on global events, even if it’s outside of their interests. As for our front end, we planned to use React (Next.js) and Vue, and for our backend, we planned to use Flask API. Finally, we

might host this application on AWS EC2, host our PostgresSQL database on AWS RDS, and utilize AWS API Gateway to set up our API endpoints.

Evaluation

1) Offline Evaluation: We might use datasets like the Microsoft News Dataset (MIND), which includes historical user interactions with the news presented to them, to train and evaluate our system. For example, we will use the first four weeks of user interactions data with news articles to train our system and then compare our recommended articles with what they clicked on in the fifth week.

2) Online Evaluation: After training our news recommender system, we will evaluate our system by the utility of our recommendations by monitoring the experience of our users. After letting our users browse multiple news articles, we want to start recommending news to them based on their user profiles. We will then evaluate our system by monitoring how much time they are spending on each news item that we recommend. We assume that our system has high utility if they spend a longer time on the news at the top of the list.

Timeline

April 3rd	Started working on the Model interface with RAG (news articles information) and User Data
April 10th	Wrap-up on model Start working on Frontend & Backend
April 24th	Wrap up frontend and backend Start putting everything together; link the model with front & back end and database
May 1st	User Study/Evaluation with the final system Start working on the Final Report
May 10th	Submit the Final Report Create the Final Presentation
May 13th	Final Presentation

Task Division

	Shuning Zhang	Yushi Li	Mayank Shetty	Rachanon Petchoo
Coordinator				✔
Backend		✔	✔	
Frontend	✔			✔
Data & ML	✔		✔	
Testing & QA	✔			✔
Deployment		✔	✔	✔
Doc & Pre	✔	✔	✔	✔