Group: Bag-of-Cats
Members: bui5, as99, romanov2, jlo10, vdara2

Progress Report (shared section)

Introduction

The main focus of our project is to give users access to the results of sentiment analysis. By combining a Chrome plug-in as the client and a dedicated RESTful web service for analyzing sentiments in Amazon product reviews, our goal is to create a user-friendly experience that improves understanding and engagement while browsing.

Workflow

Client Request: Clients initiate requests using the get_request endpoint, triggering processing.

RabbitMQ Queueing: Requests are placed in the messages queue in RabbitMQ for later analysis.

Web Pages Parser: The parser retrieves requests from the queue, inserts a row into the PostgreSQL database with a client_id and 'in progress' status. It loads the requested web page, extracting relevant data for analysis.

Analyzer Processing: The analyzer gets parsed text from the Parser, performs the sentiment analysis, and updates the database with the result.

Client Result Request: Clients request the analysis result using the get_result endpoint. The web service will return the result of analysis as soon as it becomes available in the database.

Challenges:

As discussed with the course staff, we had to pivot from doing seller analysis to doing product analysis due to complications with scrapping the former. But by force of recognizing the problem early on we were able to avoid any significant setbacks.

Working with Manifest v3 for the extension has proven to limit the amount of data we are allowed to access, so finding ways around the restrictions has been a significant challenge.

Progress Report (Catherine Bui - bui5)

Accomplishments

Finished Tasks:

- Researched about Chrome plugins due to a lack of familiarity and implemented a basic plugin to test if it was working.
- Added a user interface to the Chrome plugin that includes the extension name and icon at the top followed by placeholders for the product name and sentiment analysis scores.

Group: Bag-of-Cats
Members: bui5, as99, romanov2, jlo10, vdara2

    - Manifest.json contains some basic information about the extension, such as which HTML file and JavaScript file is being used as well as what the extension's icon will be.
    - Content.js is the JavaScript file that currently only has an alert indicating when the extension is working.
    - Popup.html is the HTML file that contains the main structure of the user interface.
    - Popup.css is the CSS file associated with popup.html that provides the styling for the HTML.
- Implemented a basic sentiment analysis function using NLTK and TextBlob.
    - The python file tests whether the review is in English or in a different language. If it is in English, the sample text will be analyzed with the SentimentIntensityAnalyzer and its scores will be printed; the scores are formatted as neg, neu, pos, and compound where the highest score between neg, neu, and pos indicates the sentiment. Compound is the aggregate score that ranges from -1 to 1 where -1 is most negative, 0 is neutral, and 1 is most positive.

Next Steps

Pending Tasks:

- Modifying the Chrome plugin UI so that it now displays information for only product reviews since the team got permission to change our focus to product reviews instead of seller reviews
- Merging the UI with Alyosha's part to replace the placeholders
    - This will allow the UI to display information that is updated accordingly to the product page
- Making the UI more dynamic as opposed to static as it is now
    - The button to close the extension needs to be working
- Improving the sentiment analysis functionality
    - Needs to be better for different languages

Challenges

- Sentiment analysis function currently does not perform well when it comes to different speaking languages, so other alternatives will need to be looked into

Group: Bag-of-Cats
Members: bui5, as99, romanov2, jlo10, vdara2

Progress Report (Aleksandr Stepenko - as99)

Accomplishments

Finished Tasks:

- Server Code for Web Service:

Completed the development of the server code for the RESTful web service, establishing the foundation for robust communication between the client (Chrome plug-in) and the server.

- RabbitMQ and PostgreSQL Setup:

Successfully configured RabbitMQ, providing seamless integration for queuing requests within the application workflow.

- Established a PostgreSQL database, laying the groundwork for persistent storage of analysis requests and results.

- Implemented a sentiment analyzer utilizing the NLTK library, contributing a vital component to the analysis pipeline.

- Made significant progress on a BERT-based solution for sentiment analysis, not ready yet.

Next Steps

Pending Tasks:

- Complete of BERT-based Sentiment Analyzer:

Continue working on the BERT-based sentiment analyzer to enhance the accuracy and depth of sentiment analysis.

- Improve Server Code:

It will help enhance overall system architecture and performance, ensuring a reliable and efficient platform for sentiment analysis.

- Functional Testing:

Conduct comprehensive functional testing to ensure all components work as expected.

Challenges

- BERT models are computationally expensive and require significant resources, especially for training. Fine-tuning on specific tasks can also be resource-intensive.
- Design a scalable, modular, and maintainable architecture for client-server application.

Group: Bag-of-Cats
Members: bui5, as99, romanov2, jlo10, vdara2

Progress Report (Alyosha Romanov romanov2)

Finished Tasks:

Created an amazon web scraper that parses a listing's reviews, returning all the necessary data in a JSON format. By scraping the DOM directly, we can avoid using Amazon's pricey API access, but we are unfortunately limited to 10 pages of 10 reviews. The list of reviews generated will be passed on for sentiment analysis. The scrapper is a fork of a previously broken project but with significant modifications to work with the new amazon website.

Started work on the extension and created a minimum viable product that displays an extension window with the product title and 3 hard coded sentiments loaded in from a file to simulate the data received from the server. Created both the background and foreground scripts, as well as established communication between the two to facilitate the displaying of relevant information.

Next Steps:

Fix bugs in the scrapper, for example there is currently a bug with accessing the "next page" which is critical for getting all 100 reviews necessary for sentiment analysis.

The extension still needs to talk to the server to get the sentiment analysis, so working on all the API access will be a significant next step.

Progress Report (Josephine Lo - jlo10)

Finished Tasks:
Began building a parser using BeautifulSoup to retrieve all Amazon seller reviews, but stopped to switch to the retrieval of Amazon product reviews. Looked into the issues Alyosha encountered from building the Amazon web scraper. Looked into Naive Bayes implementation of sentiment analysis using NLTK by understanding and familiarizing myself with the natural language toolkit using a sample movie review dataset. I then further investigated the features of NLTK's pre-trained sentiment analyzer.

Next Steps:
Continue working on issues that arise with the Amazon web scraper, such as accessing sufficient amounts of reviews. Continue flushing out the Naive Bayes implementation of sentiment analysis using NLTK.

Progress Report (Vikram dara - vdara2)

Finished Tasks:
Vikram has not contributed to the project.