OpinionNet App

CMPT 475/6
Fall 2022

1 Introduction

Your team has been contracted to develop an application to allow the student body at Marist to issue opinions about issues of their interest. This application allows anyone with access to see what are the current issues that people are bringing up and up-vote or down-vote any issue.

The requirements provided below are very high-level on purpose. Part of your responsibility is to create detailed requirements for the project by submitting questions or meeting with the client (Prof. Arias). You'll have to make certain architecture and design decisions. When you do, make sure that you keep the client informed and update your requirements document.

2 High Level Requirements

2.1 User Interface

This type of application is more useful a a smartphone app. If you feel like it would be too difficult for you to implement it as a smartphone application, you can instead implement it as a web-based application. However, the technical merits of such an interface would be less. One alternative is to create a web-based interface that also works with a smartphone.

Your interface should allow a user to:

1. Create an account (this will be separate from the Marist login because it is too cumbersome to integrate with Marist authentication). You should let users use another authentication as a way of registering and accessing the app, i.e. Google, Facebook, etc.
2. Manage their account.
3. Add a new opinion to the system. When a new opinion is added, existing opinions are matched against this new one to make sure there is no duplication. To match opinions, a natural language analysis needs to be performed to find potential matches.
4. List opinions by popularity. Display also sentiment information next to the opinion.
5. Allow to up-vote or down-vote any opinion. Users can either up-vote or down-vote an opinion, but not both. Opinions can only be either up-voted or down-voted once by each user. Votes are anonymous.
6. A previous vote to an opinion can be modified or removed by the owner.
7. Opinions expire after a specified number of days after the last vote. The number of days is configurable by a property on the server. The default is 90 days.

2.2 API

Your system should collect statistics about the opinions in the system, and should provide a RESTful API that allows for querying such statistics.

2.3 Performance

No screen in your interface should take more than 3 seconds to render the data. Your universe of users is Marist's student body. However, you should account for the possibility of expanding it and making the application available in other schools around the nation.

2.4 Technology

You are free to select your technology stack.

You will be provided virtual machines that you can use to install and develop your application. If you prefer, you can get your own cloud machines, e.g. Google Cloud, and use those.

3 Project Management

I recommend that you find a project management tool online that you can use to track your project plan, e.g., Trello, ZenHub, etc.

Track your code in GitHub and share your repository with me.

4 Resources

The following are some resources that your can use. It is not an exhaustive list, and you are not required to use any of these:

- Google Natural Language AI
- IBM Natural Language Processing
- Open API Specification
- Trello