Twitch Sentiment Analysis

Arthur Shir
Arthur.shir@gmail.com
4/24/2018

Introduction:

The goal of this project was to create an application that would connect to a streamer's channel and analyze the top 10 friendliest viewers in a Twitch channel (as well as the top 10 angriest viewers). It utilizes the Twitch Chatbot API to access stream chats, and the IBM's Watson Tone Analyzation API for tonal recognition.

Goals for this project:

○ To display technical achievement.
○ To utilize the Twitch API in an interesting and unique way.
○ To create a proof a concept of a useful business benefit.

Project Architecture:
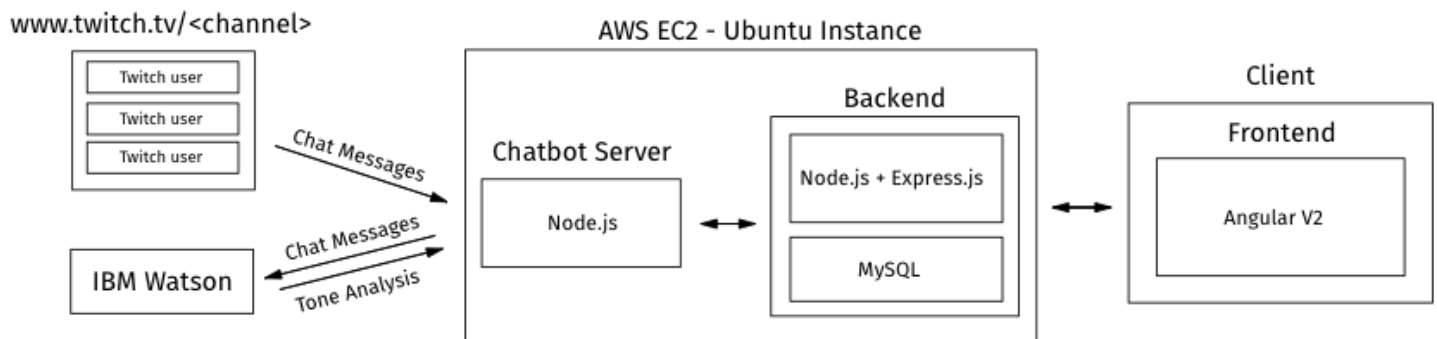
Technologies Used:

○ Angular V2, Node.js, Express.js, MySQL, Knex, tmi.js, watson-developer-cloud.js, request.js, AWS EC2

Project Flow Overview:

1. A Twitch viewer sends chat. (NOTE: For this demo, please send a chat in the form of [<username>] <message>. This is done to allow for testing data on the real life Twitch stream.)
2. The chat server picks up the chat and sends to IBM's Watson Tone Analyzer.
3. The chat server sends over the resulting analysis to the website backend.
4. The resulting rankings of the friendliest and meanest viewers (ranked by largest number of either friendly or mean chat messages from viewers) are reflected on the Angular frontend website.

Background:

I tried to think up of some kind of interesting application that extended a little further beyond a chatbot moderator / shell. As I looked into the Twitch Chatbot API and chat messages, I realized there could be something interesting in combining the Twitch Chatbot API with other interesting API's and cloud services that are out there. So, I ended up with an idea creating a sentiment analysis of a Twitch user's viewers.
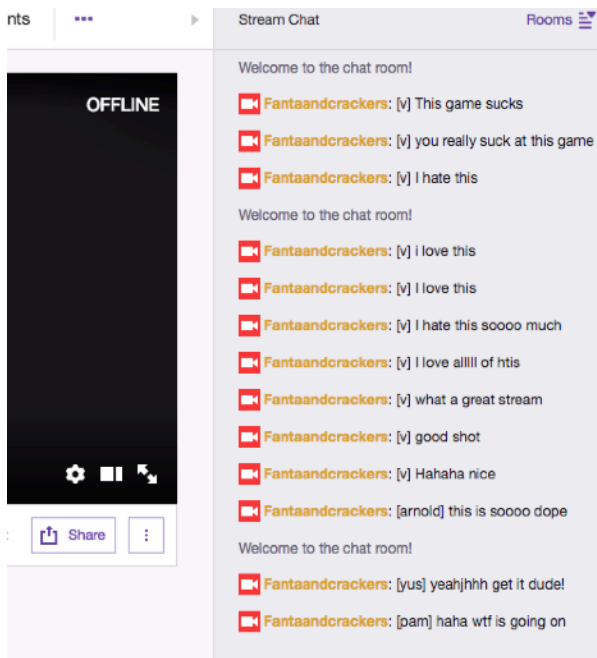
Analysis:

The most engaged viewers grouped by most the angry, the most joyful, the most fearful, and etc could potentially be very interesting for the streamers to analyze. From this information, streamers could potentially change their streaming style or methods of engagement with their viewers.

One application could be that streamers could utilize an application like this to pinpoint what viewers react the most strongly to- whether it be in a positive or negative way, and to potentially change their streaming style if it is relevant. Streamers may also be able to utilize these "top ten" lists to keep certain engaged viewers in mind when planning gift or reward drops.

Would I would do with more time to expand on this project:

- ○ Gain first hand experiences in streaming as well as do research to better understand streamer to viewer relationships and utilize this knowledge to drive feature changes.
- ○ Explore different ways to measure the top ten "most angry" or top ten "most joyful" viewers.
- ○ Explore different ways to show graphical data of viewers grouped by sentiment
- ○ Possibly store chat messages and show the most frequent messages that have the highest tonal scores.
- ○ Refactor and modify the project to be scalable and production ready.

Demo Images: