

CS415 Project 3 Proposal

Jacob Barkovitch
Binghamton University
Vestal, New York, USA
jbarkov1@binghamton.edu

Guy Ben-Yishai
Binghamton University
Vestal, New York, USA
gbenyis1@binghamton.edu

Alan Bixby
Binghamton University
Vestal, New York, USA
abixby1@binghamton.edu

Jacob Coddington
Binghamton University
Vestal, New York, USA
jcoddin1@binghamton.edu

Ryan Geary
Binghamton University
Vestal, New York, USA
rgeary1@binghamton.edu

Joseph Lieberman
Binghamton University
Vestal, New York, USA
jliebe12@binghamton.edu

1 INTRODUCTION

For our project 3, we plan to implement a basic REST API to automatically generate graphs that represent single team and team head-to-head sentiment and post frequencies and use quantitative measurements from these graphs to make and test a prediction against a given sportsbet bookmaker. At the most basic implementation, this will be an API that returns json and image data; if time permits a HTML front end may be created for user-convenience.

1.1 Research Questions

For our final project, our group will continue to focus on the following research questions:

- (1) Can sentiment analysis and posting frequency of social media data be used to predict the expected winner of a game?
- (2) Can sentiment analysis and posting frequency of social media data be used to identify team rivalries and classify reactions to game outcomes?

In project 2, we created a time series of each NFL game, tracking the polarity and post frequency, but our analysis was largely qualitative. For the first research question, comparing aggregate difference of the moving averages will be able to quantitatively identify which team was more optimistic leading into the game commence time and by how much. For predictions, this analysis will focus on the trends *prior* to the game commence time, for reactions, the analysis will focus on the trends *during*, and *after* the game.

Currently we plan a conservative REST API interface that would be hosted locally on the VM using Flask for the webserver and API routes, and pymongo for database access. As a stretch goal, this may be recreated as a React dashboard, but pragmatically, this is unlikely.

In project 2 and as seen in Figure 1 we created an automated way to graph team versus team game window analysis which can be done for both posting frequency and sentiment analysis. As the variable parameters, one could provide a team name and see a .png output of the team sentiment and frequencies over time for all games, or reconstruct head-to-head data if it exists. Furthermore, one could modify the bin width (30 minute interval) and the before and after day thresholds could be another parameter within the REST API interface, as currently our paper only utilizes 2 days before, 2 days after, at 90 minute bins. Given the nature of these graphs, this could be done for Reddit and Twitter, and for both frequency of posting and sentiment analysis. For preliminary graphs, please refer to the *plots* branch of our [implementation repository](#).

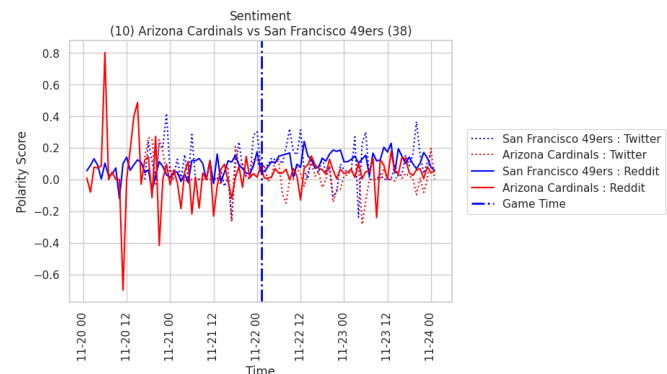


Figure 1: Example graph generation

We could also integrate into the tool a sampling method that lets a user preview a random set of tweets or comments from within a sampling period to qualitatively validate the sentiment scores provided by TextBlob.

2 TOOLS FOR RESEARCH

The tools we intend to use for research are:

- Python
- Flask
- Numpy/Pandas/Matplotlib/Seaborn
- pymongo (MongoDB connection)
- Docker

2.1 Dynamic Parameters

The types of graphs focused on will be the following:

- (1) Individual team sentiment and post frequency
- (2) Team vs team sentiment and post frequency surrounding a game day

And the parameters within the graphs that can be varied will be:

- Team Name
- x-Axis Limit
- Data Source
- Binning Frequency
- Moving Average Width
- Use Aggregate Difference

And these parameters may be used against the Odds API to see if it correlates or goes against a given bookmaker's odds, and may let you make a bet and see what the payout would've been.

3 ACCESS

Our interactive tool will be accessible on localhost, and we expect instructors to SSH into the VM while on VPN to run and access the demonstration. If only a REST API is used, SSH tunnelling into the server may be required so a browser or HTTP request manager such as Postman can be used to make the web requests.