Jonathan Wiseman

February 17, 2020

CS495

Milestone 1 Report

**Goals:**

1. Dataset collection
2. Dataset cleaning
3. Selection of sentence embedding algorithm

**State of the Project:** <https://github.com/jonwiseman/DraftSense>

A screenshot of a cell phone

Description automatically generatedThe fundamental goal of the project remains unchanged: produce a sentiment analyzer trained on Reddit comments for gauging reactions to NFL draft picks. The overall structure of the project is shown in figure 1:

**Figure 1: New Project Flow**

Once all comments have been scraped and separated into their respective threads, they will be consolidated into one dataset file (tentatively called comments.json). After the comments have been consolidated, *clean\_comments.py* will remove URLs, links to other subreddits and users, emojis, parentheses, and brackets. Finally, the script *label\_comments.py* allows a user to label comments. After labeling has been complete, the project will focus on model creation.

**Progress**

1. Dataset collection

The script *scrape\_comments.py* accomplishes this objective. Using PRAW, *scrape\_comments.py* collects all top- and second-level comments for a given player’s draft thread. If the player is not found in the thread, then the program raises a PlayerError. If the player is found, then all top- and second-level comments are scraped. There is no cleaning performed in this step, so comments consisting of URL links, auto-moderated comments, and deleted comments are all present; instead, the task of cleaning is left to *clean\_comments.py*.

Running *scrape\_comments.py* for the target 12 players yields the following results:

* Baker Mayfield: 1381 comments
* Mitchell Trubisky: 1945 comments
* Daniel Jones: 3165 comments
* Kyler Murray: 1571 comments
* Lamar Jackson: 889 comments
* Dwayne Haskins: 948 comments
* DeShaun Watson: 910 comments
* Sam Darnold: 637 comments
* DeShone Kizer: 589 comments
* Josh Rosen: 673 comments
* Josh Allen: 766 comments
* Patrick Mahomes: 959 comments

This yields a total of 14434 comments that will have to be labeled. For each player, a .json file with the following fields was created: comment\_id, post\_id, comment. The comment\_id is the unique comment identifier, the post\_id is the parent post (i.e. the draft reaction thread), and the comment is the text data.

1. Dataset cleaning

The script *clean\_comments.py* cleans each comment in the consolidated comments.json file in the following way: removing URLs, removing newline characters, removing emojis, removing parentheses, removing brackets, and removing other extraneous characters. Punctuation and capitalization are left unchanged, as are the original word forms (i.e. no stemming or lemmatization has been performed). As it functions now, cleaning gets rid of scraping artifacts rather than significantly changing the text data. Keeping the text largely unchanged will help in sentence embedding, since linguistic structure will be largely preserved.

1. Selection of sentence embedding algorithm

I will be using both Google’s BERT and Sent2Vec to train embeddings. The final project will decide which one specifically to use, based on which algorithm produces the highest accuracy and fastest speed.

1. Other

The script *consolidate\_comments.py* is an addition to the pipeline that simplifies dataset creation. Rather than trying to label 12 players in 12 separate .json files, I instead gather them all into one .json file.

The script *label\_comments.*py is the first attempt at a script to label the dataset. It functions through simple command line prompts to the user (such as fetch a comment and label a comment). The labeling scheme is as follows:

* -1: unlabeled
* 1: positive
* 2: negative
* 3: joke or meme
* 4: junk/irrelevant

I do my best to avoid labeling errors by having two running records of the dataset: the comments.json file and a separate DataFrame that is pickled after each labeling session. If ever the two have a comment with labeling inconsistency, an error is thrown with the index noted. This script will be the primary workhorse for the next milestone.