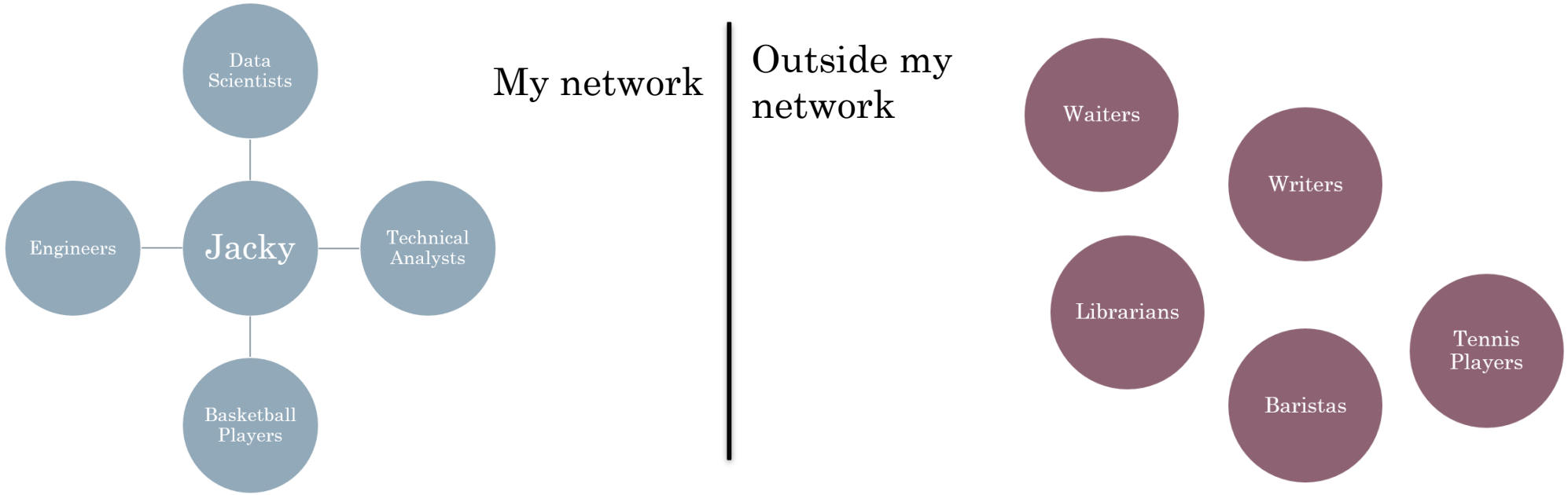
Network Recommendation Engine

Jacky Zhao, April 2018

Problem

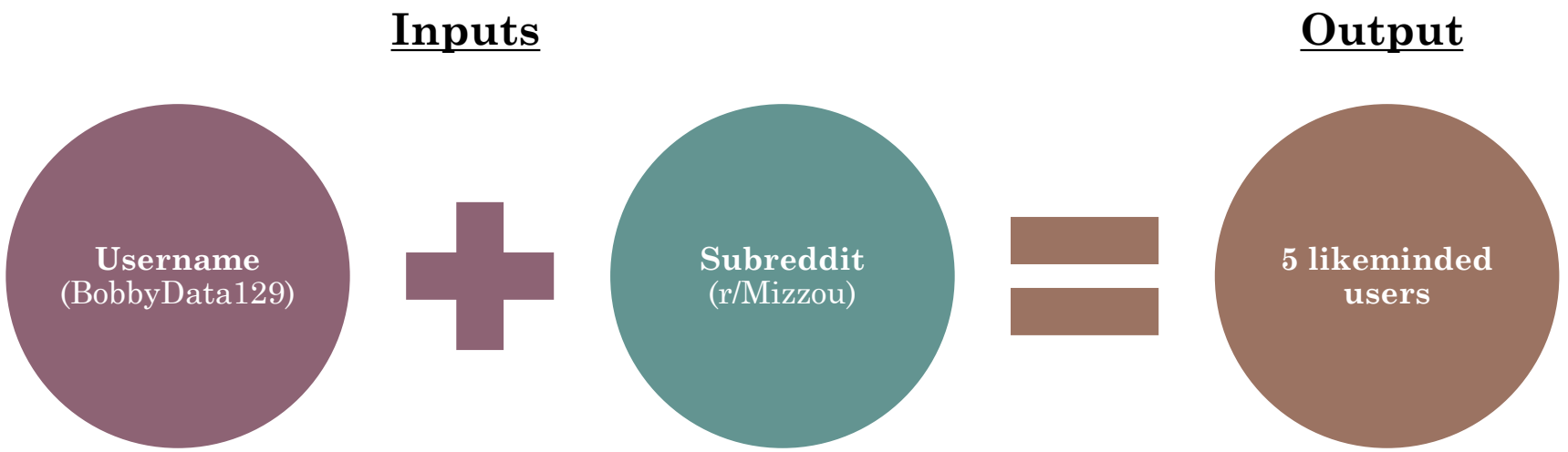
- People tend to self-identify and stay within their own social circle of friends and colleagues.
- They are unwittingly excluding others with different labels from their network.



Proposed solution

- Recommendation engine API that refers you to 5 likeminded people based on your reddit comments and subreddit of choice.

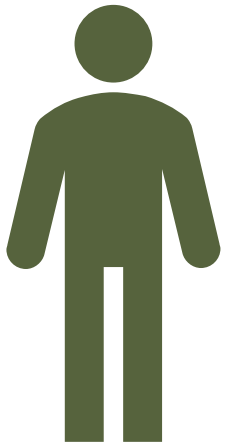
  (https://github.com/iamjz/matcher)



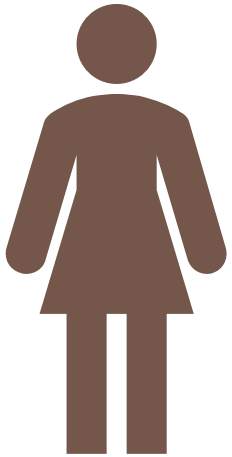
What makes this unique?

- Matches individuals based on talking points instead of personality questionnaires.
  - **Traditional:** OkCupid, eHarmony, Elite Singles, Match.com
- Under traditional networking methods, these 2 would have a low chance of meeting.

- **Likes college basketball**
- **Low-carb diet**
- **Likes modern art**



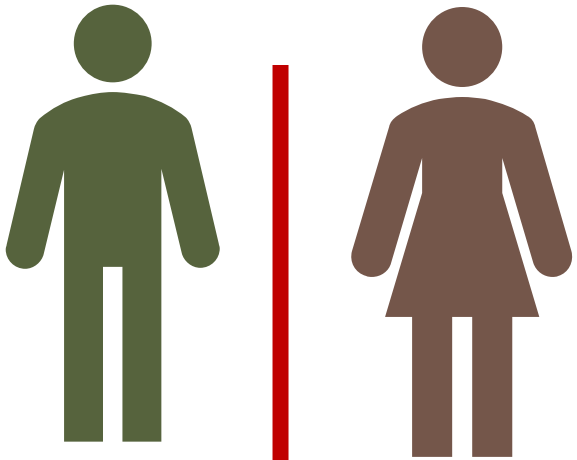
Bob



Sally

- **Watches football**
- **Tries out different restaurants**
- **Does graffiti**

- Likes college basketball
- Low-carb diet
- Likes modern paintings



Bob

Sally

- Watches football
- Tries out different restaurants
- Does graffiti

Sports	Food	Art
Coaches	Smoothie	Colors
Recruits	Kale	Vibrant
Wins	Bacon	Masterpiece
Preseason	Delicious	Paint
Drafts		

Implementation

Gather

- Reddit API
- Python library (PRAW)
- Data cleansing
- Storage

Recommend

- TF-IDF
- Porter Stemmer
- Stop-word removal
- Cosine similarity
- Outputs top 5 based on similarity score

Validate

- Common subreddits
- Similar flairs
- Hierarchical document clustering

Implementation: Gather

- Reddit RESTful API (https://www.reddit.com/wiki/api):
  - Requires a Reddit developer account for an API Client ID and Client Secret Key
  - Free usage – up to 60 requests per minute
- PRAW Python Reddit API wrapper (https://praw.readthedocs.io/en/latest):
  - Only requires basic understanding of Reddit.com and Python
  - Great documentation
- Data cleansing/manipulation (helpers.py):
  - Organized API requests in functions and loops
  - Try/Catch blocks to account for suspended/banned/deleted accounts
  - Removed newlines, page-breaks, non-alphanumeric characters, and extra spaces
- Storage:
  - Only scraped a small fraction of each subreddit.
  - Stored the cleaned comments in csv files (user – comments)

Implementation: Recommend

- TF-IDF (Term frequency – Inverse document frequency):
  - TF (Term frequency): Measures how frequently a term occurs in a document.
    - Increasing by the number of times a word appears in a document.
  - IDF (Inverse document frequency): Measures how import a term is.
    - Decreasing by the number of times a word appears in the entire corpus.
- Porter stemming algorithm:
  - Reduces a word back to its word stem or root form.
  - Improves performance in text analytics by categorizing various similar words.
- Remove stop-words:
  - Filter out common words without any significant meaning (the, is, a, that, at, which…)
  - Combined 2 stop-words dictionaries for extra filters.
- Cosine similarity:
  - Measures the “distance” by finding the cosine angle between 2 vectors (documents)
  - Catches the direction of the vectors better = semantic meaning
  - Euclidean distance not as powerful for high dimensional data

Example:

In [11]:
```
# NOTE: I'm displaying the top 6 usernames who are similar.
# I chose 6 because if the username is contained in my scraped dataset,
# then that username will always show up as most similar.
# But we really only care about the top 5 usernames.

matches = findMatches("Max_W_", "mizzou", r)
```

```
--- Retrieved 288 corpuses for r/ mizzou
------ Stemming the words
------ Retrieved 1875 comments for: Max_W_
------ Stemming the words
--- Creating Tfidf vectorizor...
--- Fitting the matrix...
--- Retrieved 288 corpuses for r/ mizzou
...
...
Username: Max_W_ | Score: 0.989111748103
============================================================
...
...
Username: BrettGilpin | Score: 0.566956453617
============================================================
...
...
Username: EveryTrueSon | Score: 0.555547201932
============================================================
...
...
Username: Apatches | Score: 0.550465800003
============================================================
...
...
Username: BransonBombshell | Score: 0.550037564134
============================================================
...
...
Username: SovreignTripod | Score: 0.545633770494
============================================================
```

View code @ https://github.com/iamjz/matcher/

Implementation: Validate

Validations: Common Subreddits

Examples

```
one, two = commonSubredditCounts("Max_W_", "BrettGilpin", r)

# Output to CSV files for visualization.
one.to_csv("frontend/common_sub1.csv", index = False)
two.to_csv("frontend/common_sub2.csv", index = False)
```

```
Retrieved 2420 comments for user: Max_W_
Retrieved 2838 comments for user: BrettGilpin
31 common subreddits found...
```

Max_W_

id	value
adviceanimals	30
android	64
askmen	1
askreddit	371
atheism	2
aww	5
baseball	2
cfb	1
funny	43
gifs	22
iama	5
imgoingtohellforthis	2
justiceporn	1
mapporn	18
mizzou	34
morbidreality	6
movies	28
news	16
nfl	21
nottheonion	1

BrettGilpin

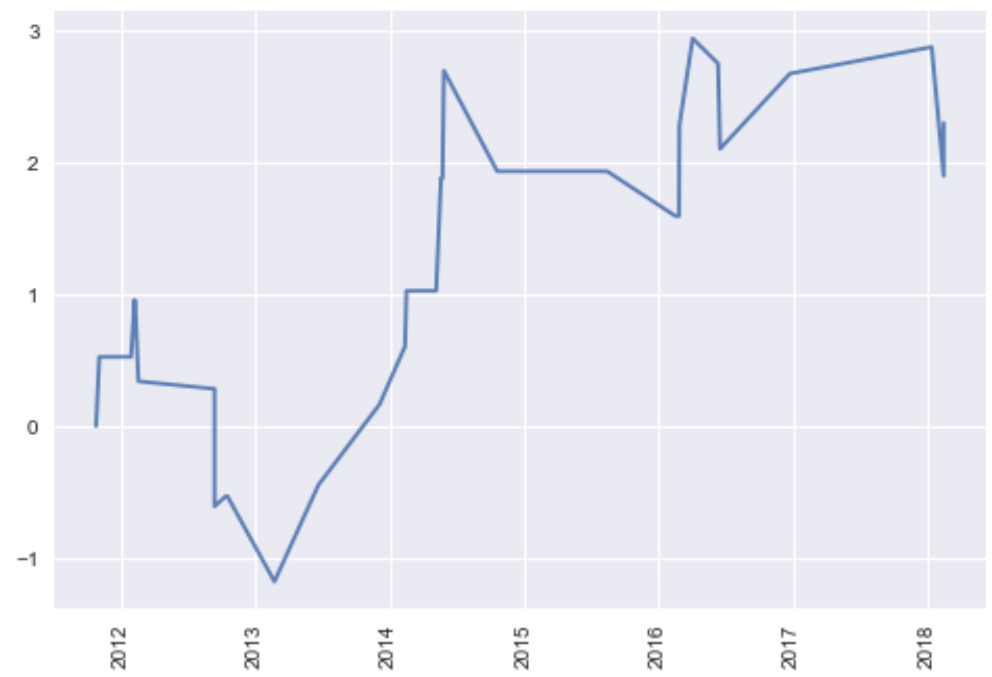
id	value
adviceanimals	8
android	73
askmen	151
askreddit	82
atheism	1
aww	39
baseball	17
cfb	560
funny	1
gifs	1
iama	16
imgoingtohellforthis	5
justiceporn	2
mapporn	1
mizzou	74
morbidreality	1
movies	1
news	22
nfl	4
nottheonion	18

View code @ https://github.com/iamjz/matcher/

Validations: Subreddit Sentiments

```
plotCumSentiment("Max_W_", "mizzou", r)
```

```
Retrieved 2413 comments for user: Max_W_
```




```
plotCumSentiment("BrettGilpin", "mizzou", r)
```

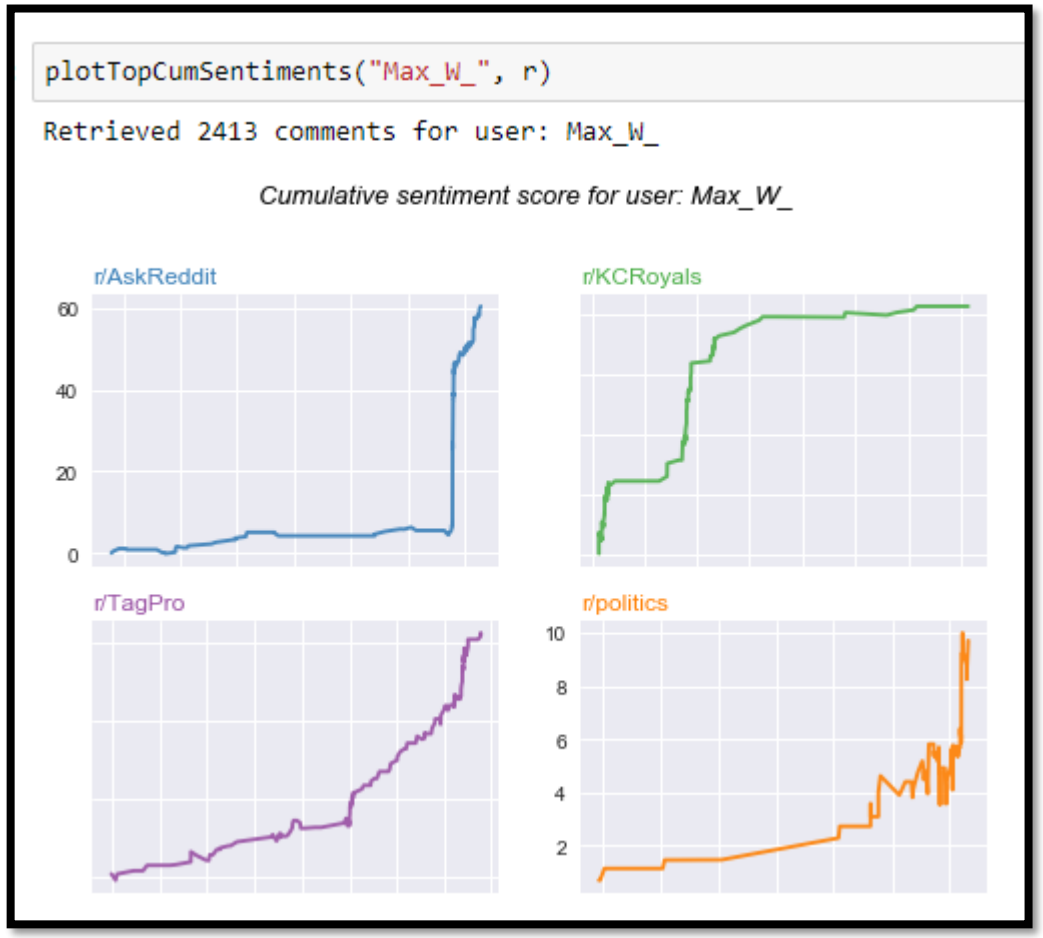
```
Retrieved 2838 comments for user: BrettGilpin
```

View code @ https://github.com/iamjz/matcher/

Validations: Top Subreddit Sentiments

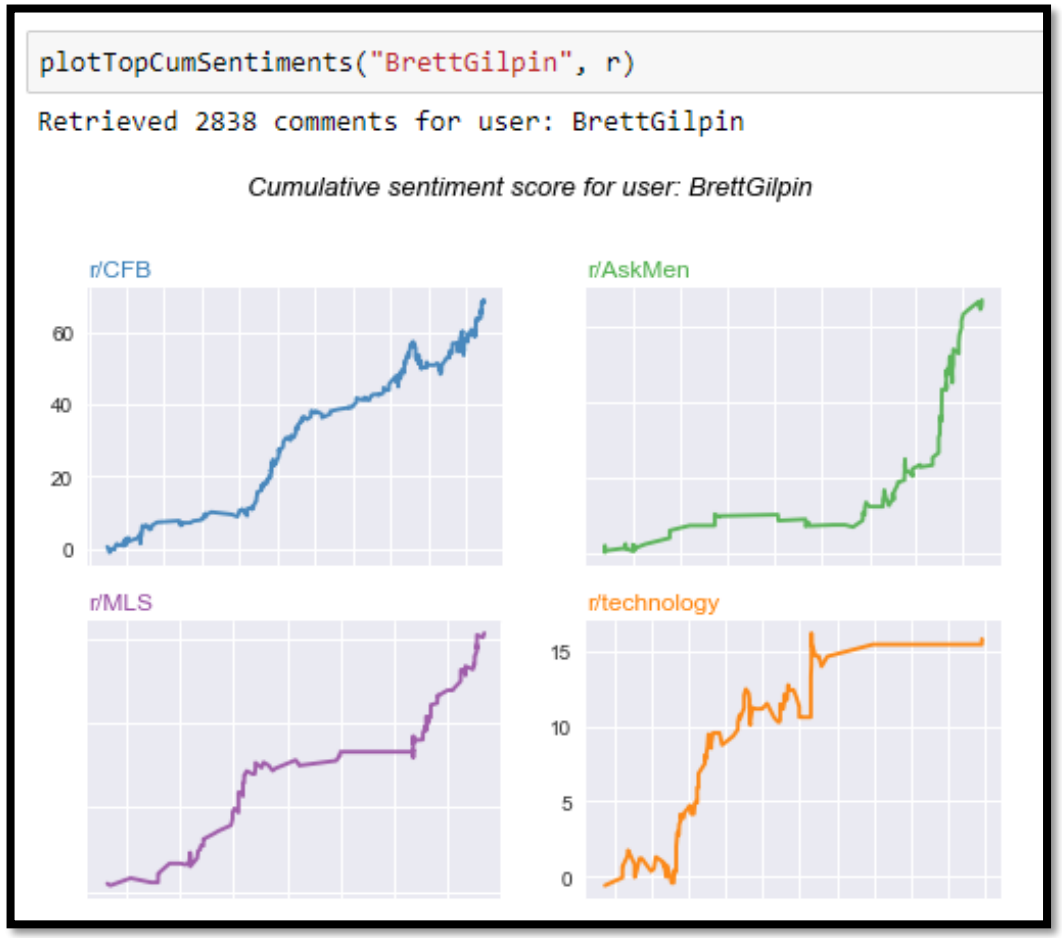
```
plotTopCumSentiments("Max_W_", r)
```

```
Retrieved 2413 comments for user: Max_W_
```




```
plotTopCumSentiments("BrettGilpin", r)
```

```
Retrieved 2838 comments for user: BrettGilpin
```



View code @ https://github.com/iamjz/matcher/

Validations: User Flairs

```
getFlairs("Max_W_", r)
```

```
Retrieved 2420 comments for user: Max_W_
{'Chiefs',
 'Honor 7X / Asus Nexus 7 / Moto e4+ ',
 'Honor 7X, Blue',
 'Kansas City Royals',
 'MaxW // Force Sensitive',
 'Missouri',
 'Nexus7 FHD',
 'Royals are clutch!'}
```

```
getFlairs("BrettGilpin", r)
```

```
Retrieved 2838 comments for user: BrettGilpin
{'"The Struggle"',
 '59s',
 'Awesome Sauce',
 'Chiefs',
 'Computer Eng Alumni',
 'FC Kansas City',
 'HTC One M8 - AT&T',
 'Kansas City Royals',
 'Male',
 'Missouri',
 'Missouri Tigers',
 'Missouri Tigers / Dartmouth Big Green',
 'Sporting KC',
 'Sporting Kansas City',
 'ð'}
```

View code @ https://github.com/iamjz/matcher/

Validations: Common Features

```
getCommonFeatures("Max_W_", "BrettGilpin", r)

------ Retrieved 1875 comments for: Max_W_
------ Retrieved 1958 comments for: BrettGilpin
```

```
{'wa easili',
 'wa end',
 'wa entir',
 'wa fake',
 'wa free',
 'wa grab',
 'wa kind',
 'wa later',
 'wa movi',
 'wa nearli',
 'wa nice',
 'wa possibl',
 'wa pretti good',
 'wa reaction',
 'wa rock',
 'wa roll',
 'wa sad',
 'wa taken',
 'wa told',
 'wa worri',
 'wage peopl',
 'wait black',
 'wait dont',
 'want attent',
 'want dont',
 'want hi',
 'want id',
 'want make',
 'want new',
 'want someth',
 'want want',
 'wasnt wa',
 'watch watch',
 'wave',
 'way better',
 'way describ',
```

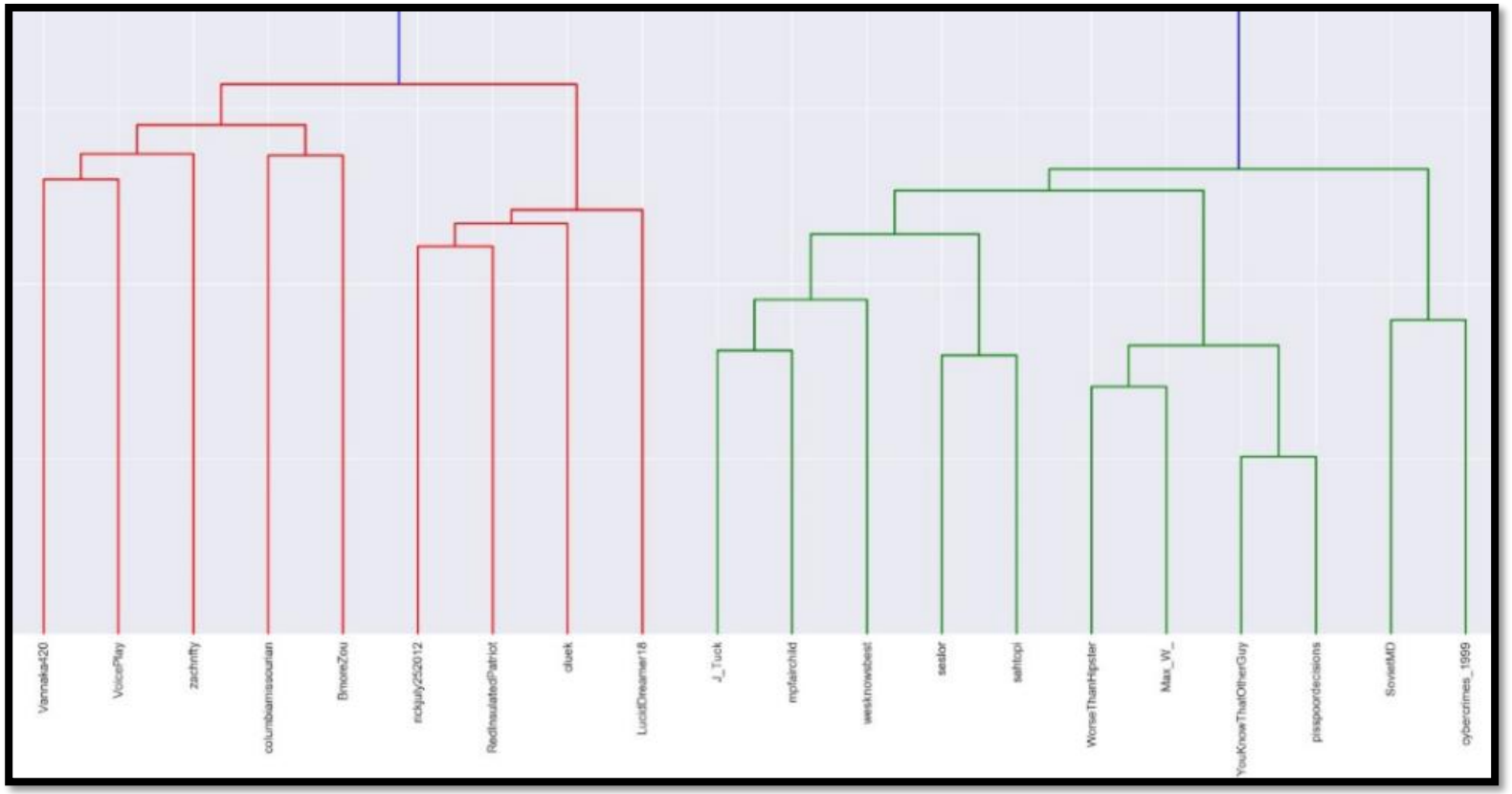
```
'way veri',
'weather',
'wed fuck',
'wendi',
'whi doe need',
'whi everi',
'whi got',
'whi need',
'whi shouldnt',
'wikipedia articl',
'willi',
'win divis',
'wine',
'wit',
'wonder thi',
'wonder whi',
'wont let',
'work att',
'work fine',
'work phone',
'work sometim',
'worn',
'wouldnt work',
'written wa',
'wrong wa',
'xda',
'ye thi',
'year colleg',
'year miss',
'year someon',
'year thi',
'yearli',
'yoga',
'youd like',
'youtub didnt',
'youv gotten',
'zoom'}
```

View code @ https://github.com/iamjz/matcher/

Validations: Hierarchical Clustering

```
buildDendo("Max_W_", "mizzou", r, 20)

--- Retrieved 288 corpuses for r/ mizzou
------ Stemming the words
------ Retrieved 1875 comments for: Max_W_
------ Stemming the words
Corpus Records: 20
--- Creating Tfidf vectorizor...
--- Fitting the matrix...
Matrix Shape: (20, 5801)
Distance Shape: (20, 20)
--- Retrieved 288 corpuses for r/ mizzou
Number of usernames: 20
...
...
Username: YouKnowThatOtherGuy | Score: 0.548391995608
============================================================
...
...
Username: WorseThanHipster | Score: 0.513130879048
============================================================
...
...
Username: pisspoordecisions | Score: 0.481092662036
============================================================
...
...
Username: seslor | Score: 0.478670825614
============================================================
...
...
Username: J_Tuck | Score: 0.443553298888
============================================================
...
...
Username: sahtopi | Score: 0.384828738617
============================================================
```



View code @ https://github.com/iamjz/matcher/

Lessons

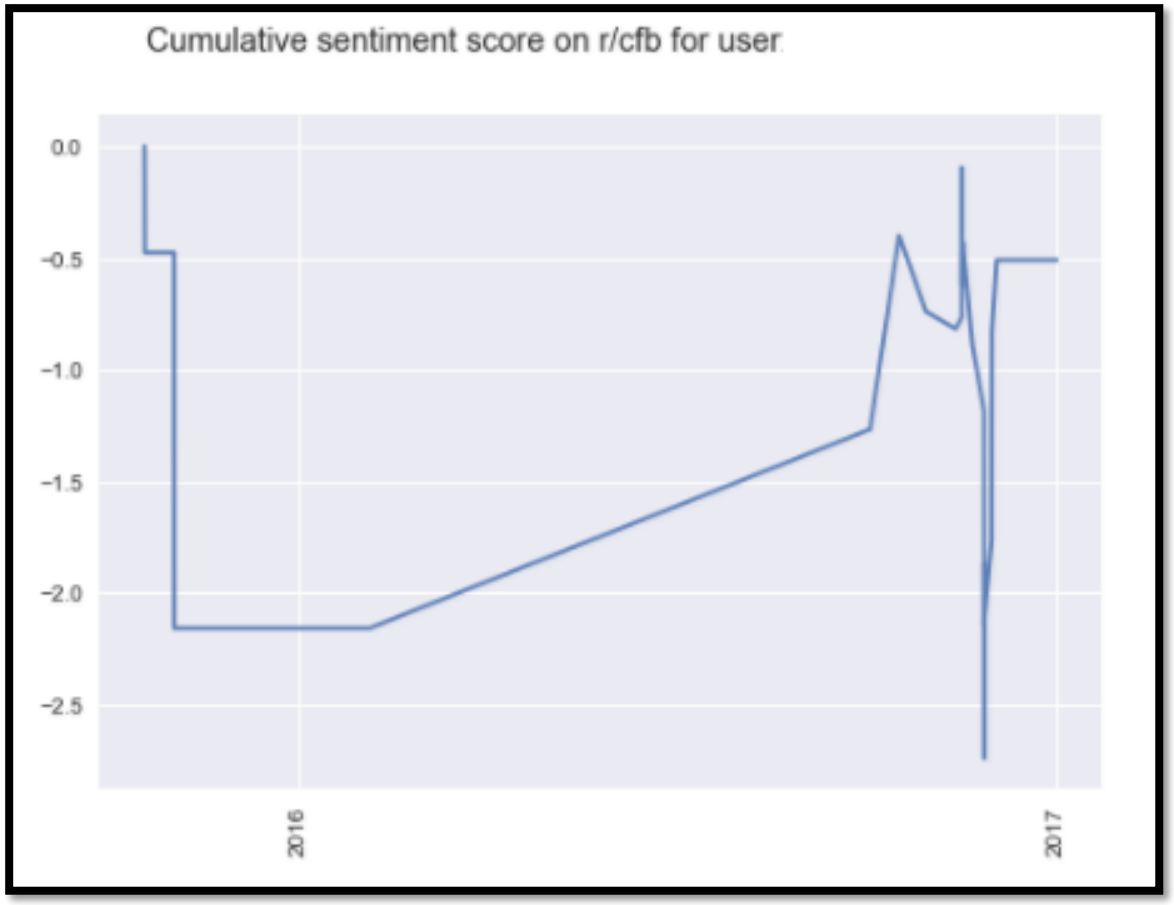
The Good

- Reddit API and PRAW
  - Great documentation
  - RESTful
  - No rate limits
  - API wrapper easy to use
- Cosine similarity on the TFIDF matrix is fast = < 5 seconds
- Recommendations made sense
  - Personal Finance ~ Financial Independence and Early Retirement

The Bad

- Volume of data
  - Unable to do this in real-time
  - Had to store the data somewhere
  - Spent a lot of time hitting the API
- Quantitative validation
  - Explore common subreddits
  - Measure sentiment
  - Ensure same clusters

Just for fun…



Sentiment of a University of Michigan fan on a college football subreddit.

* username has been hidden