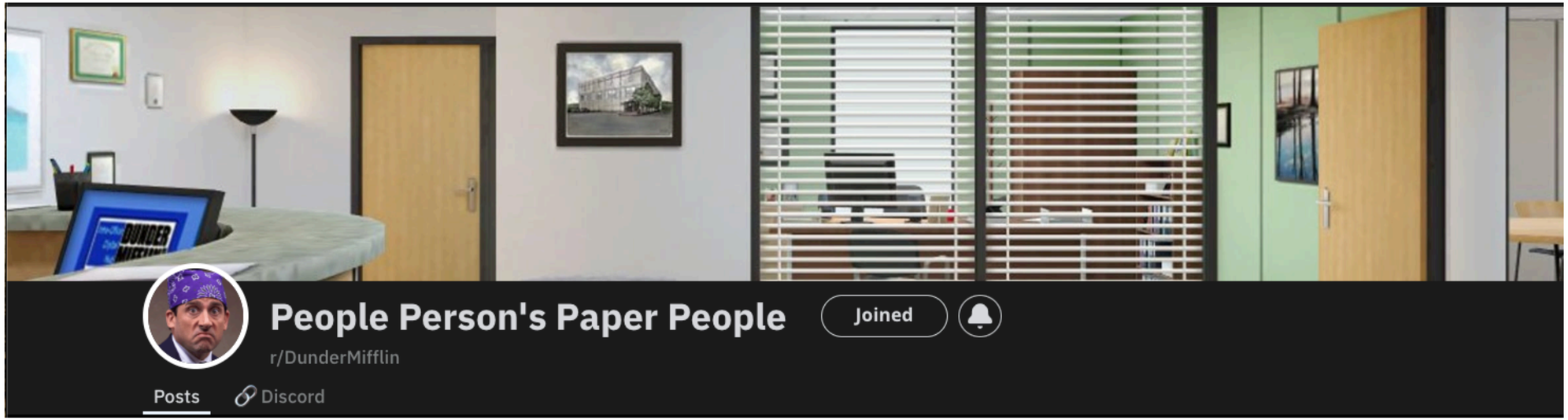
Reddit posts classification

r/DunderMifflin <> r/Office

Problem statement

Based on the title and body of the reddit post, predict if it belongs to subreddit r/DunderMifflin

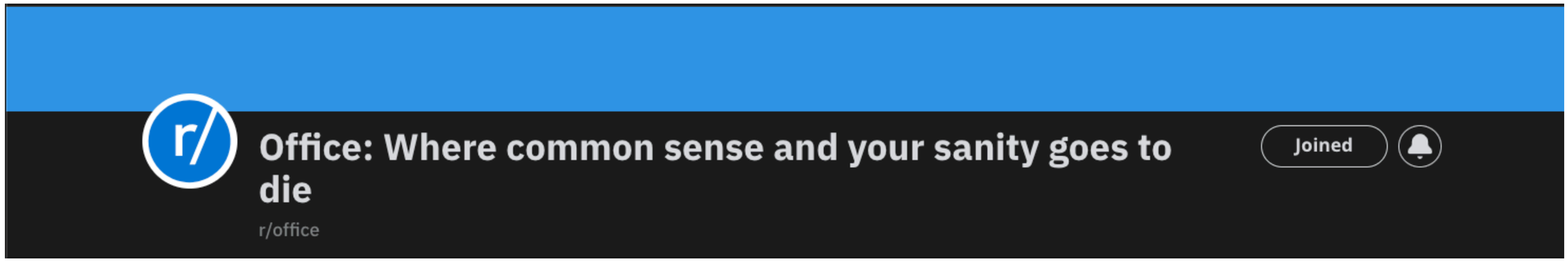


2.1m members

5 - 10 posts per day

5.6k members

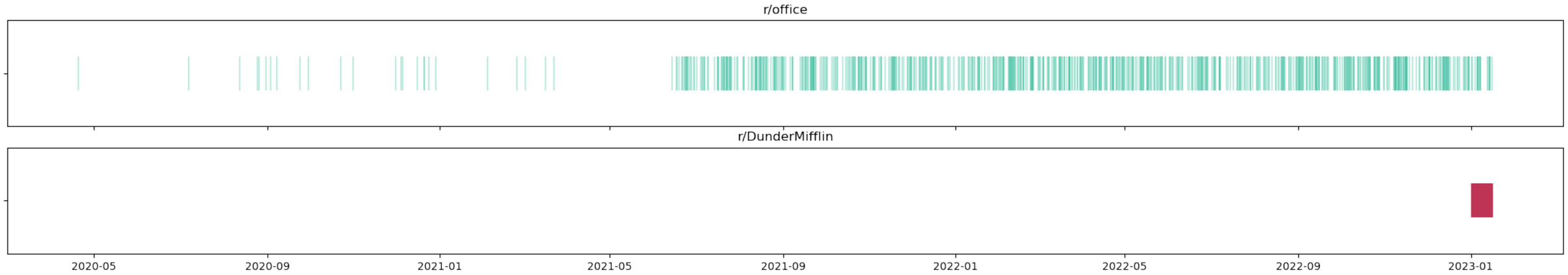
5 - 10 posts per week



Data Timeline

r/office: 793 posts

r/DunderMifflin: 993 posts

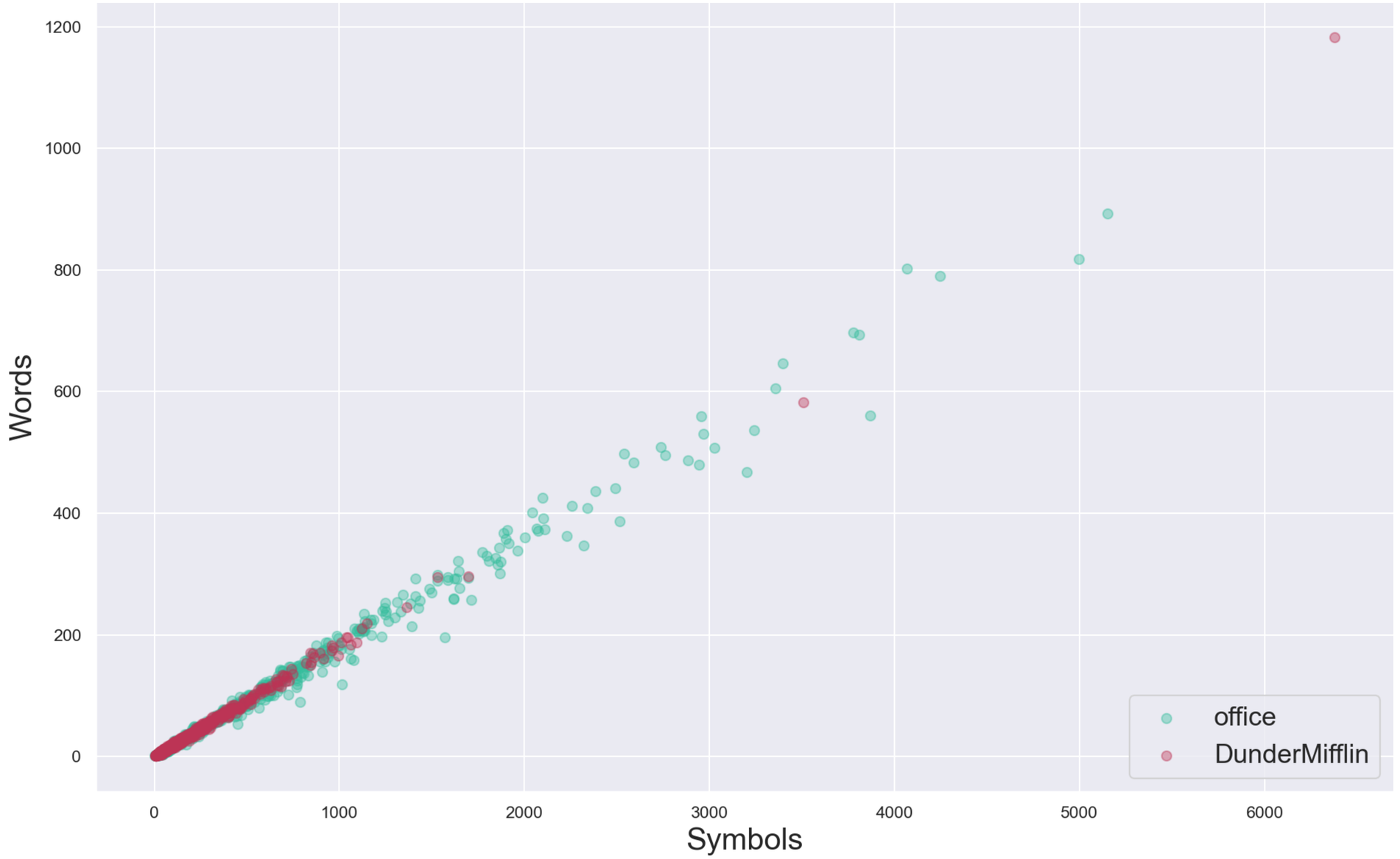


Data Timeline

r/office: 793 posts

r/DunderMifflin: 993 posts

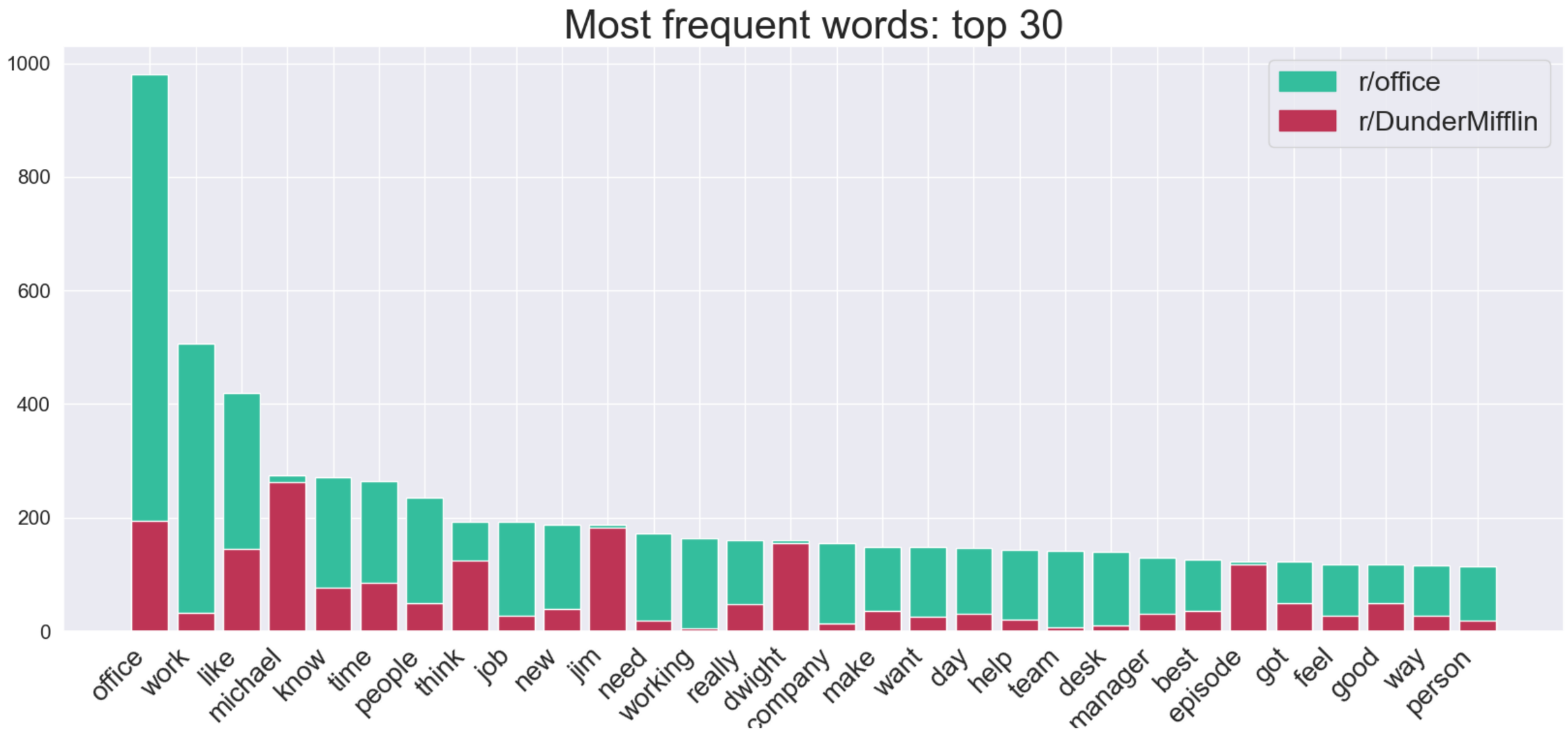
Posts length



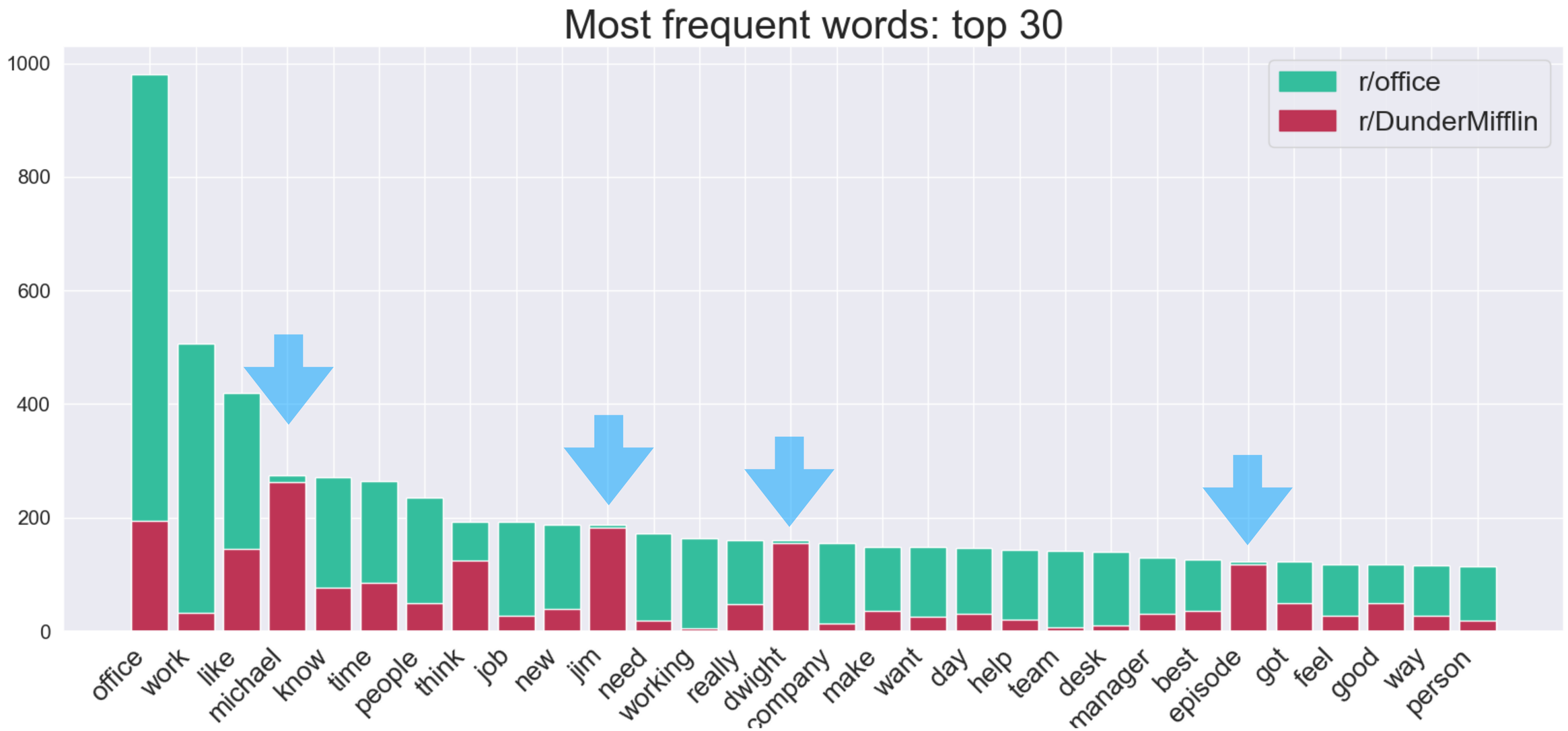
Cleaning and preprocessing

- Removing links
- Removing Special characters:
  - \n, , **, \\

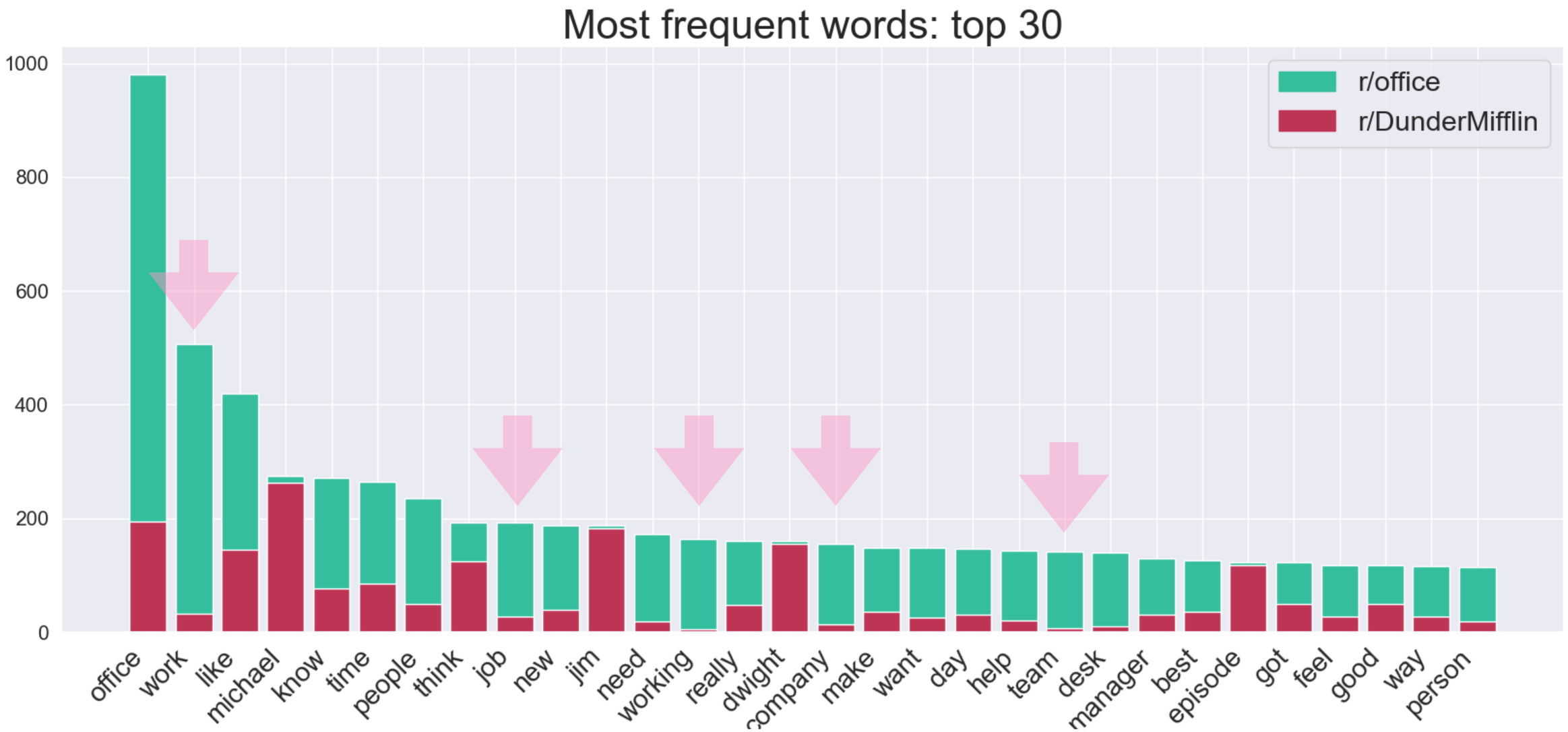
Common words



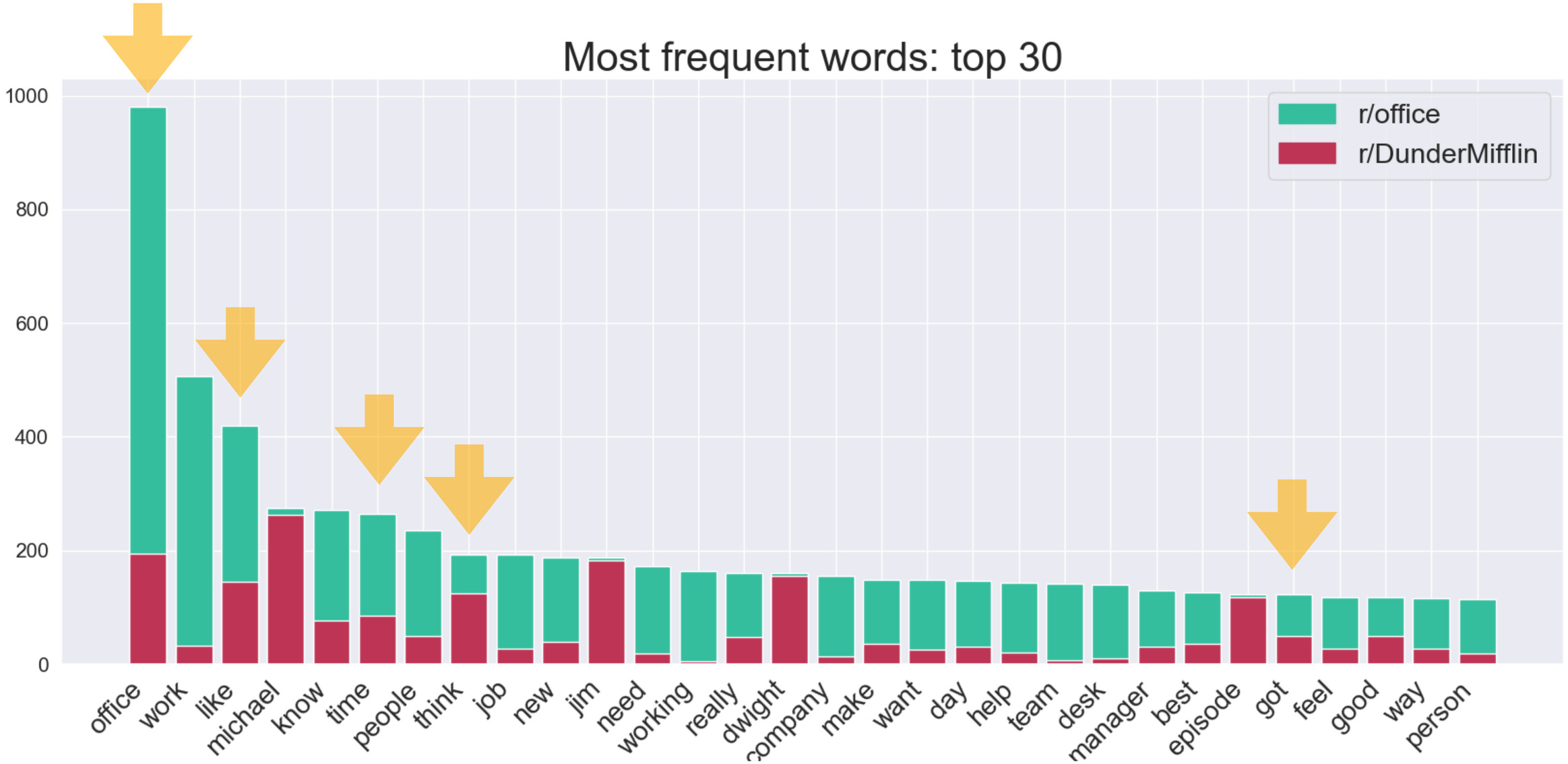
Common words



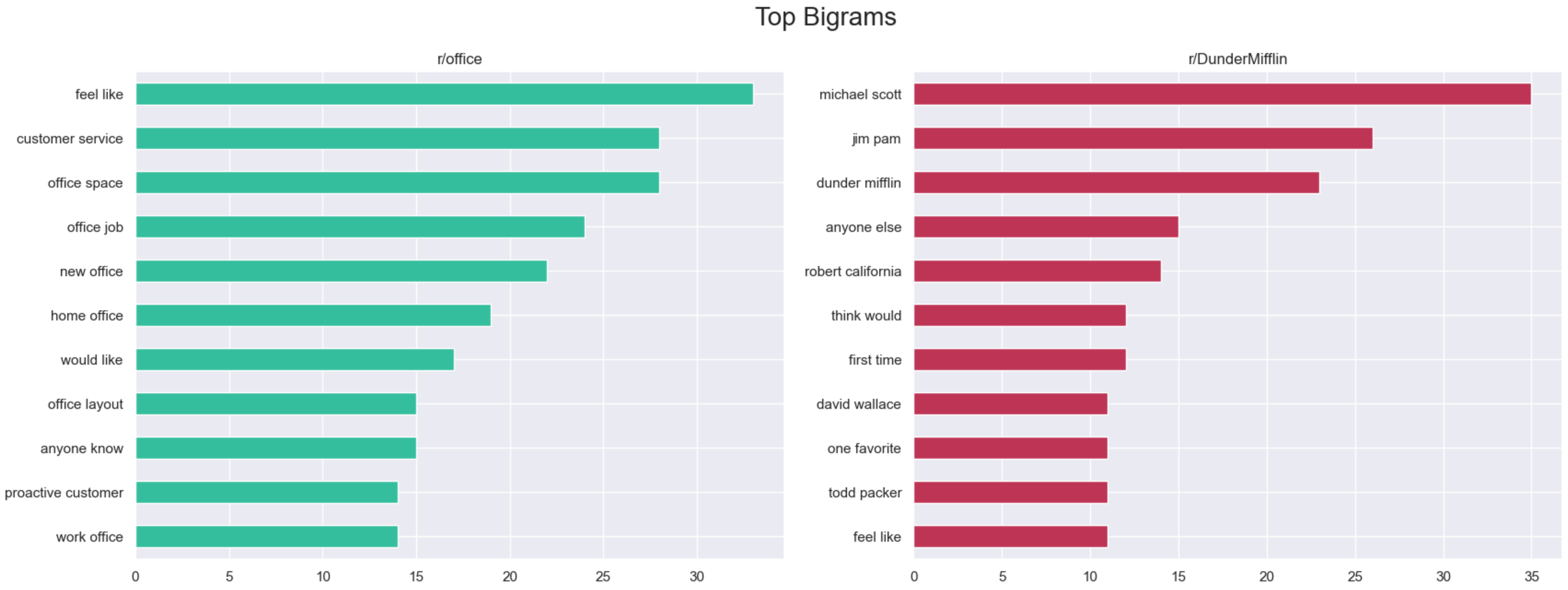
Common words



Common words



Bigrams


Top Bigrams
r/office
feel like
customer service
office space
office job
new office
home office
would like
office layout
anyone know
proactive customer
work office
0
5
10
15
20
25
30
r/DunderMifflin
michael scott
jim pam
dunder mifflin
anyone else
robert california
think would
first time
david wallace
one favorite
todd packer
feel like
0
5
10
15
20
25
30
35

Modeling

- Baseline Accuracy score: 56%

- Multinomial Naive Bayes: TF-IDF Vectorizer vs Count Vectorizer
- KNN: TF-IDF Vectorizer vs Count Vectorizer

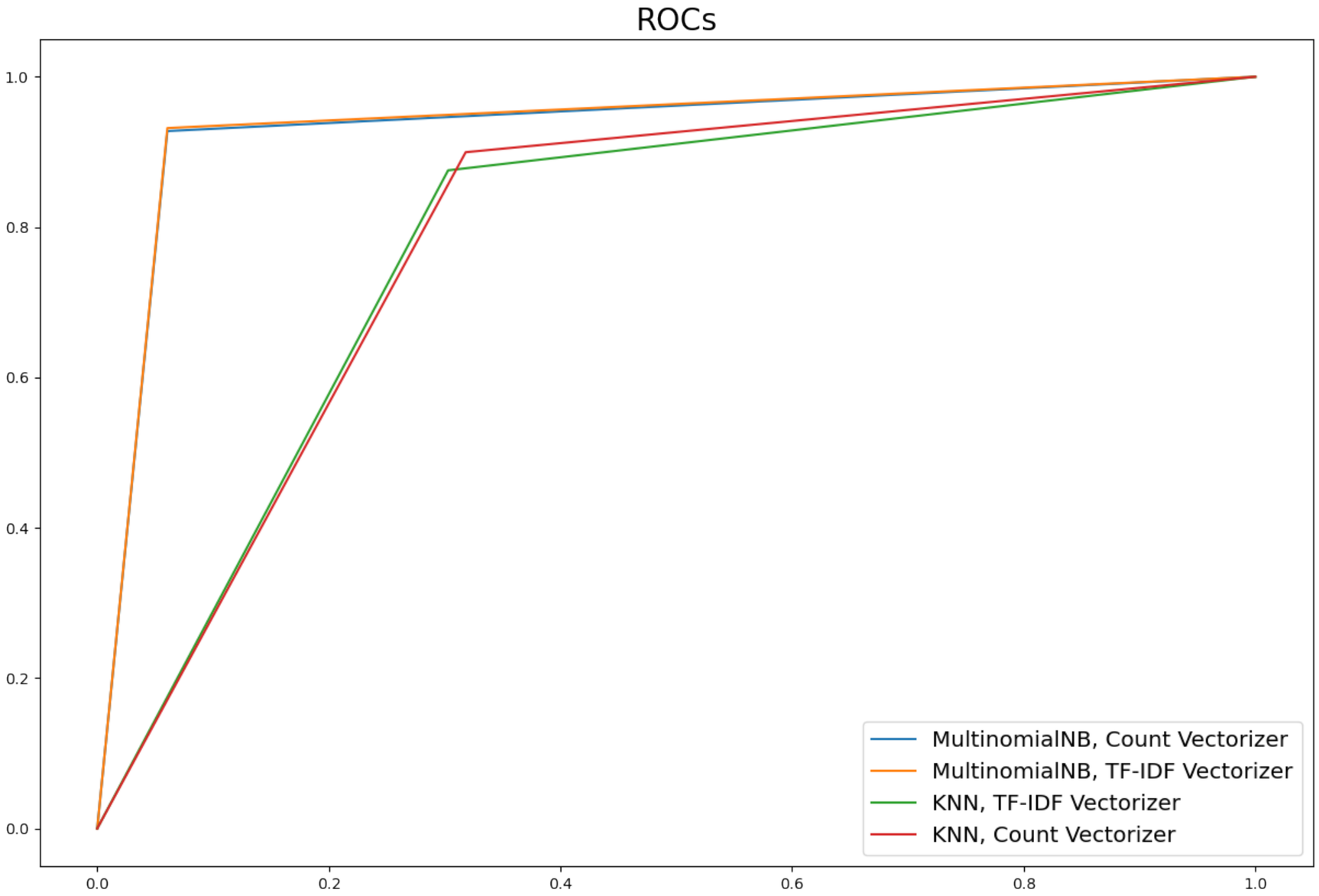
HyperParameters

Top performing Model: Multinomial Naive Bayes + TF-IDF Vectorizer

- Train score: 0.95
- Test Score: 0.94
- F1: 0.94

- max document frequency: 0.8
- max documents: 2000
- min document frequency: 2
- ngram: 1
- stop words: custom stop words
- MNB alpha: 0.2

Models comparison



Conclusions

On this project I was able to create a classification model, predicting if given post belongs to subreddit r/DunderMifflin with 94% accuracy outperforming baseline accuracy by 28%

Future performance improvements may be made by using other models, more thorough data engineering or better data cleaning.