Sarcasm Detection

“The best project EVER!”

Team Big Talk

Gennadii Styov
Jennie Lee
Michael Alaniz
Mirko Draganic
Steve Braich

Outline

- Project Overview
- Data Being Used
- Approach
- Results We Hope to Get
- Conclusion
- Questions

Project Overview

- Topic: Classifying text with software tool, a form of Natural Language Processing (NLP).
- Goal: Create a software tool that detects sarcasm in text.
- Data used: reddit posts and responses
- Tools Explored:
    - Natural Language Toolkit (NLTK) Python
    - Jupyter Notebook
    - RStudio
    - MongoDB
    - Bash
- Results so far.

The Data: An Introduction

- A repository of reddit posts and responses compiled by a team at Princeton.

- The posts organized by category

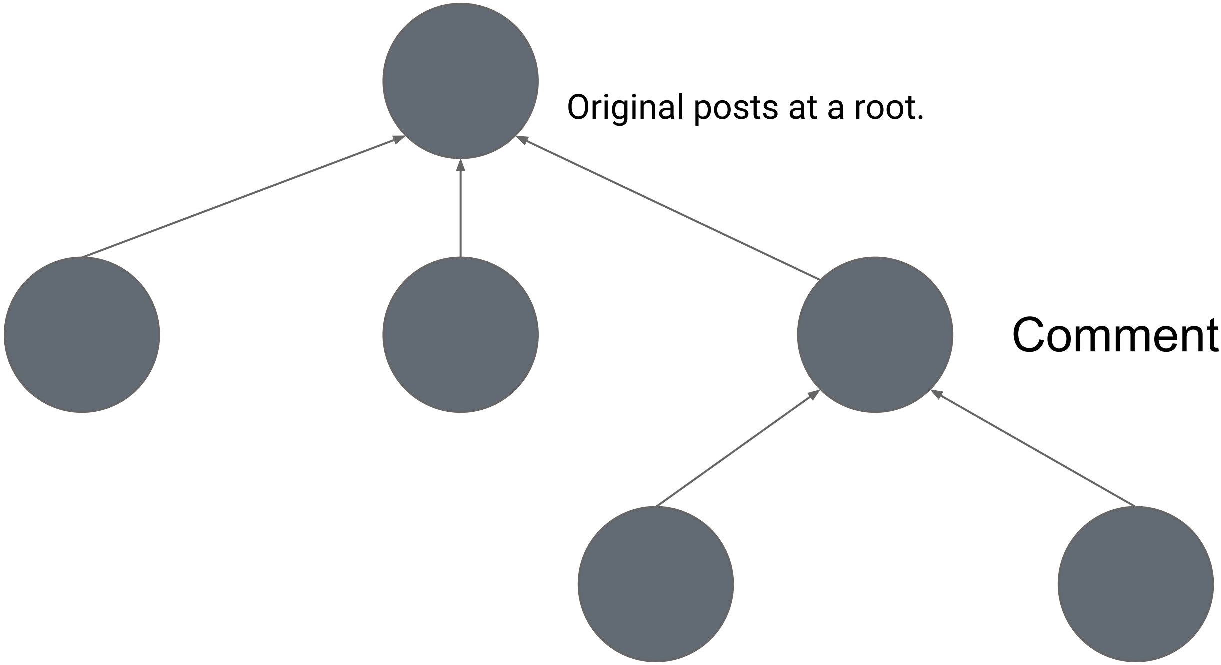
533	million total
531.7	million non-sarcastic
1.3	million sarcastic

- The authors organize the data into 3 different files.

sarc.csv:	184, 340, 693 KB
comments.json:	2, 604, 106 KB
sequences.csv:	225, 507 KB

Data found at: https://www.kaggle.com/sherinclaudia/sarcastic-comments-on-reddit#train-balanced-sarcasm.csv
Created by: https://nlp.cs.princeton.edu/

The Data: Reddit’s Forest Structure



Original posts at a root.

Comments on posts at L1.

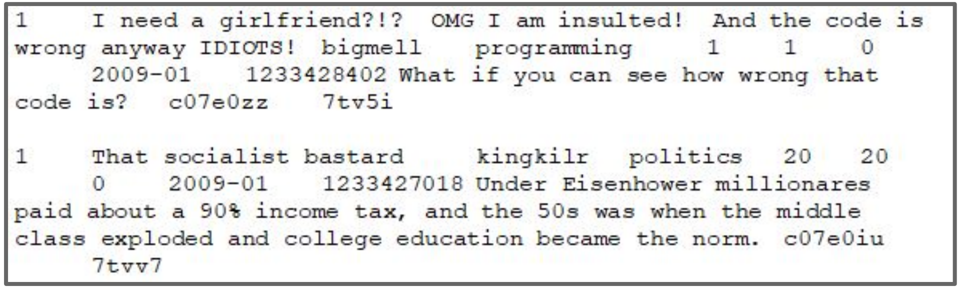
Comments on comments at L2, L3, etc…

The Data: Princeton’s Structure

sarc.csv: minimal context.

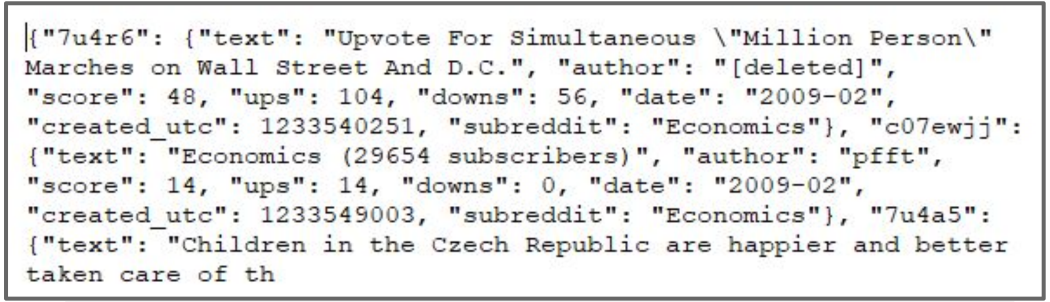
comments.json: extensive context.

sequence.csv: key to json.

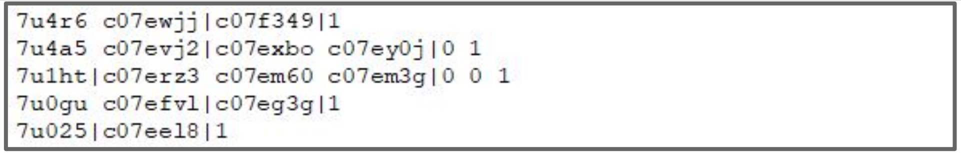


```
1     I need a girlfriend?!?  OMG I am insulted!  And the code is
wrong anyway IDIOTS!  bigmell    programming     1     1     0
      2009-01    1233428402 What if you can see how wrong that
code is?   c07e0zz    7tv5i

1     That socialist bastard     kingkilr   politics   20    20
      0     2009-01    1233427018 Under Eisenhower millionares
paid about a 90% income tax, and the 50s was when the middle
class exploded and college education became the norm.  c07e0iu
      7tvv7
```



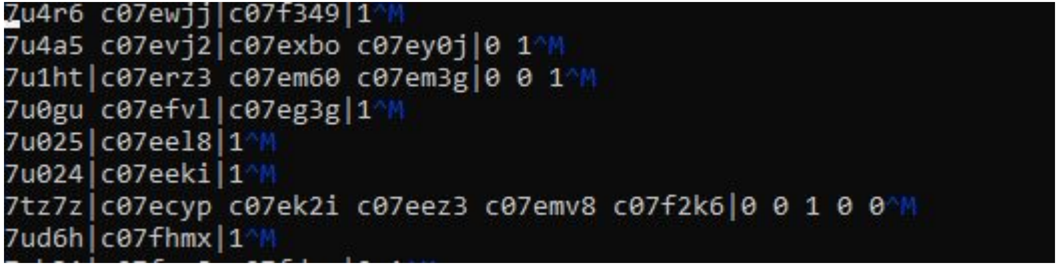
```
{"7u4r6": {"text": "Upvote For Simultaneous \"Million Person\"
Marches on Wall Street And D.C.", "author": "[deleted]",
"score": 48, "ups": 104, "downs": 56, "date": "2009-02",
"created_utc": 1233540251, "subreddit": "Economics"}, "c07ewjj":
{"text": "Economics (29654 subscribers)", "author": "pfft",
"score": 14, "ups": 14, "downs": 0, "date": "2009-02",
"created_utc": 1233549003, "subreddit": "Economics"}, "7u4a5":
{"text": "Children in the Czech Republic are happier and better
taken care of th
```



```
7u4r6 c07ewjj|c07f349|1
7u4a5 c07evj2|c07exbo c07ey0j|0 1
7u1ht|c07erz3 c07em60 c07em3g|0 0 1
7u0gu c07efvl|c07eg3g|1
7u025|c07eel8|1
```

The Data: Our Cleaning

- The sarc file was large and had to be seperated, damaging some data.
- There were lots of inconsistencies.
- Tab and newlines an early issue, quotations, and later capitalization.
- Primarily saved with grep (head, cat, Bash tools) and python.

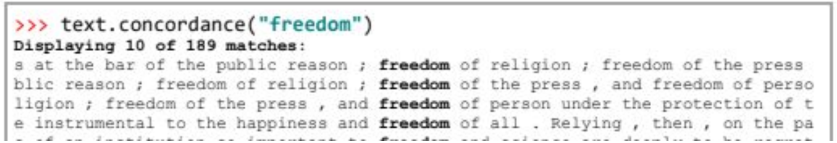


```
7u4r6 c07ewjj|c07f349|1^M
7u4a5 c07evj2|c07exbo c07ey0j|0 1^M
7u1ht|c07erz3 c07em60 c07em3g|0 0 1^M
7u0gu c07efvl|c07eg3g|1^M
7u025|c07eel8|1^M
7u024|c07eeki|1^M
7tz7z|c07ecyp c07ek2i c07eez3 c07emv8 c07f2k6|0 0 1 0 0^M
7ud6h|c07fhmx|1^M
```

- After several cleaning iterations had the following data to analyze, by number of lines:
  - Sarcastic: 1, 346, 312 lines
  - Non Sarcastic: 531, 905, 068 lines
  - Stuff we couldn’t classify: 667, 174, 383 lines

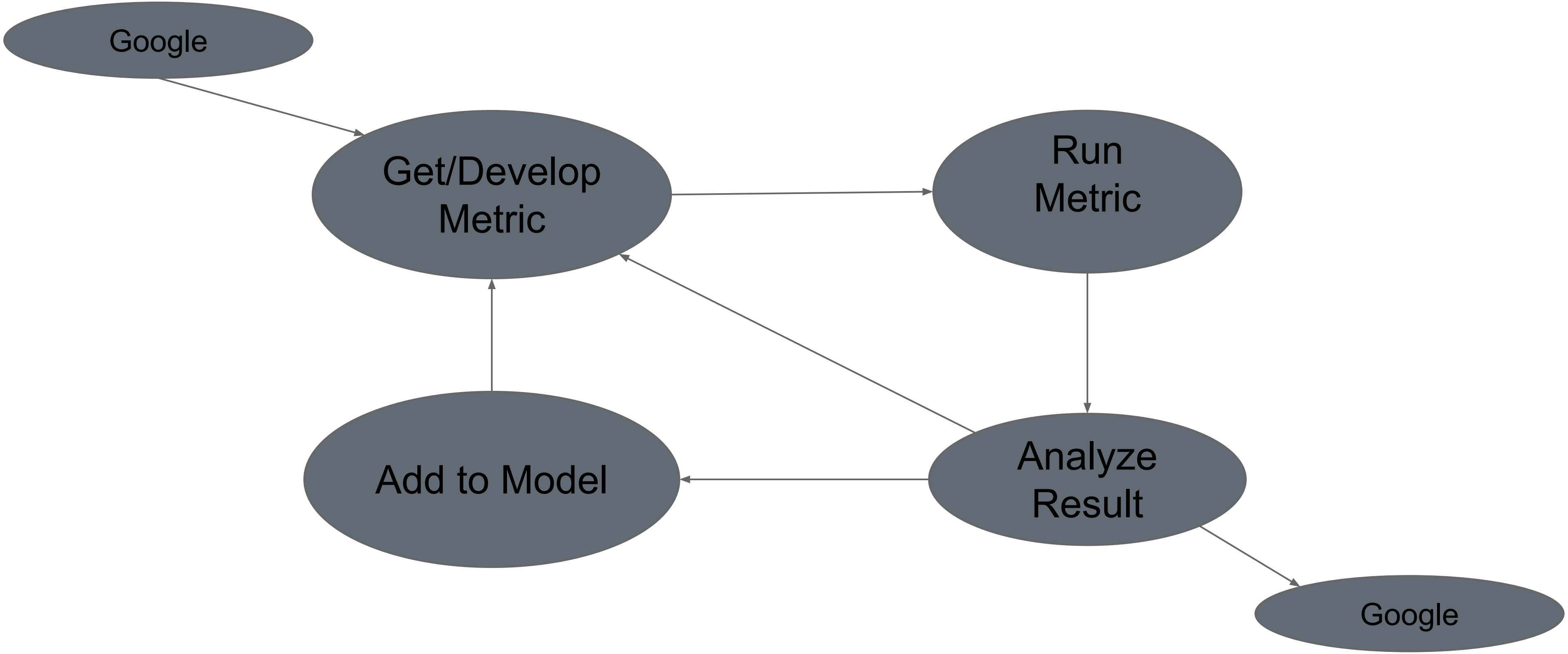
Approach: Set Up for Analysis

1. Lots of googling.
2. Found some metrics:
    a. Most frequent word, called a key.
    b. Most frequent words, around a key.
3. Found some dummy data
    a. Samples from our unprocessed data.
    b. Preprocessed samples of text from software libraries.
4. Found some language processing tools, and visualization tools.
5. Made our development environments
    a. We’re using a variety of tools, but a single set of metrics and data formats.



```
>>> text.concordance("freedom")
Displaying 10 of 189 matches:
s at the bar of the public reason ; freedom of religion ; freedom of the press
blic reason ; freedom of religion ; freedom of the press , and freedom of perso
ligion ; freedom of the press , and freedom of person under the protection of t
e instrumental to the happiness and freedom of all . Relying , then , on the pa
e of an institution so important to freedom and science are deeply to be regret
```

Approach: Iterations of Analysis



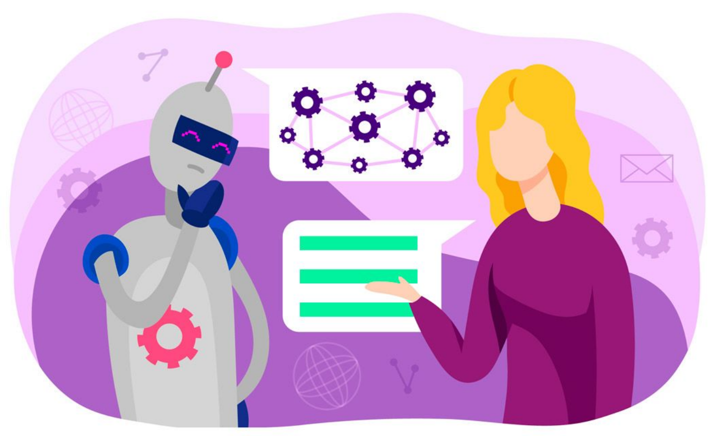
Natural Language Processing

(NLP)

What is NLP?

Natural language processing (NLP) is the technology stack used to enable computers to understand, interpret and manipulate human (natural) languages.

Natural Language Processing (NLP)

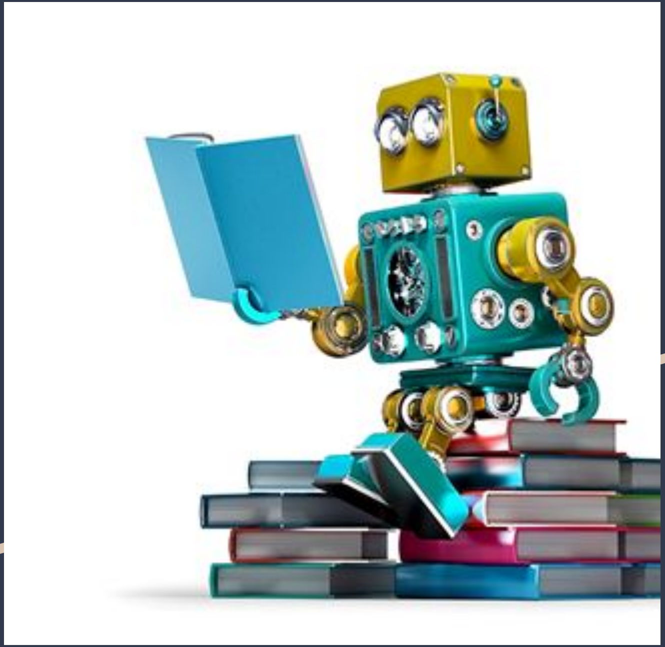


Why do we need it?

In order to build a sarcasm detector, we need to process, analyze, and ultimately build a model from corpora that has sarcastic content.

NLP Libraries

Natural Language Processing



Popular NLP Libraries used with Python:

- Natural Language Toolkit (NLTK)
- Scikit-learn
- TextBlob
- spaCy

NLTK Library (Python)

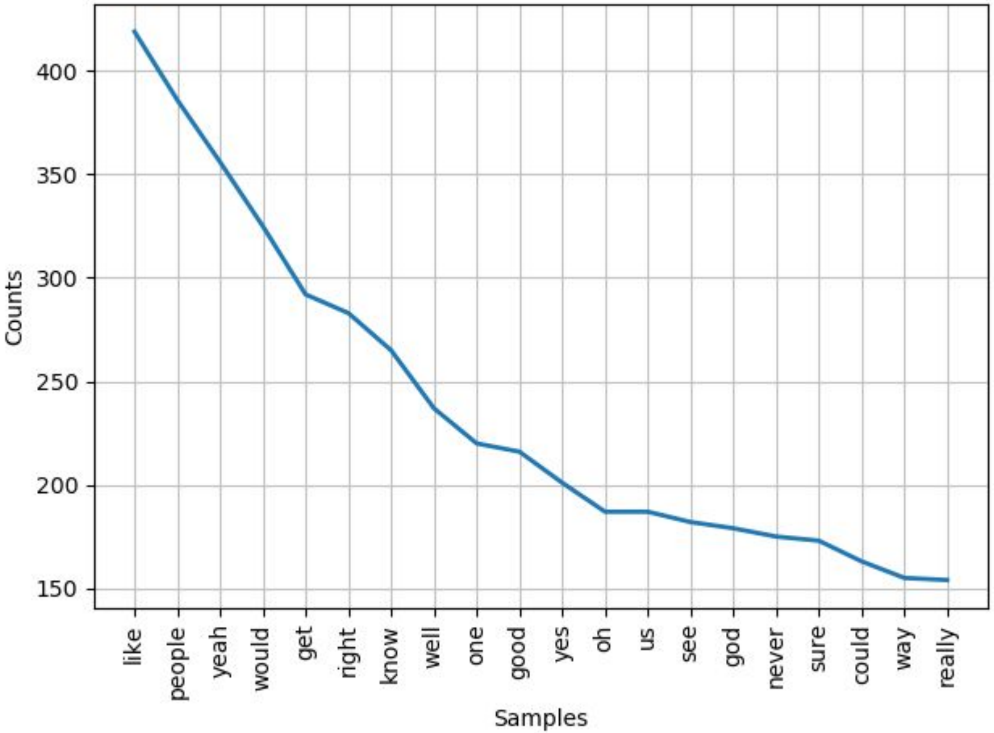
Natural Language Toolkit

- NLTK is an open source suite of natural language processing libraries for Python
- It is freely available , easy to install , and has a large community contributing to its development.
- Jupyter Notebook

  https://github.com/steve3p0/cs510ds/blob/master/nlp_analytics.ipynb

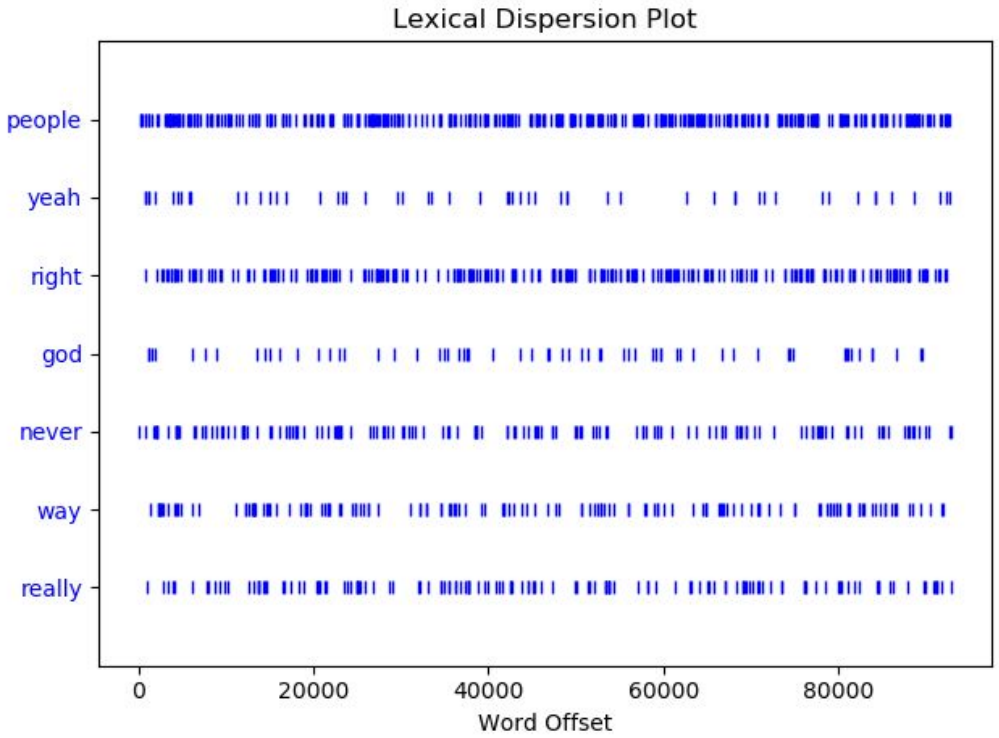
NLTK: Frequency Distribution

Top 20 Most Frequent Sarcastic Comments



NLTK: Lexical Dispersion Plot

How to find a word's importance weighed by its lexical dispersion in a corpus



Analytics with R

- R is a programming language and free software environment for statistical computing and graphics

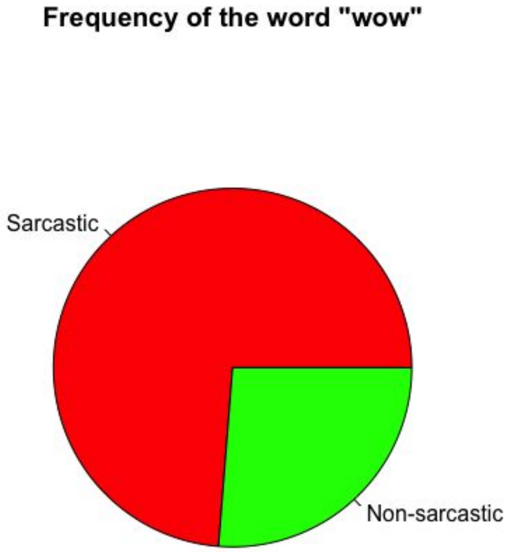
- “RStudio” is an IDE for R

- 25,934 sarcastic comments

- 25,000 non-sarcastic comments

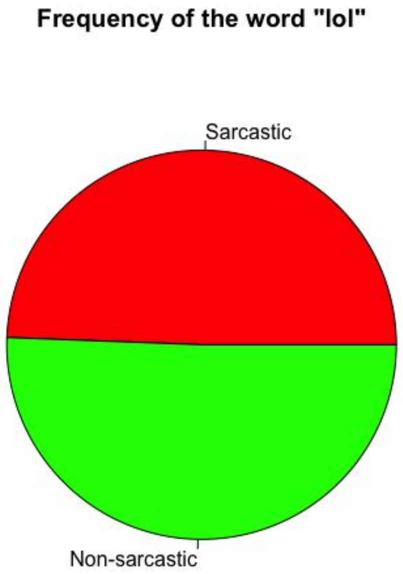
Analyzing the word “wow”

- 0.39% of sarcastic comments
- 0.14% of non-sarcastic comments



Analyzing the word “lol”

- 0.3% of sarcastic comments
- 0.31% of non-sarcastic comments



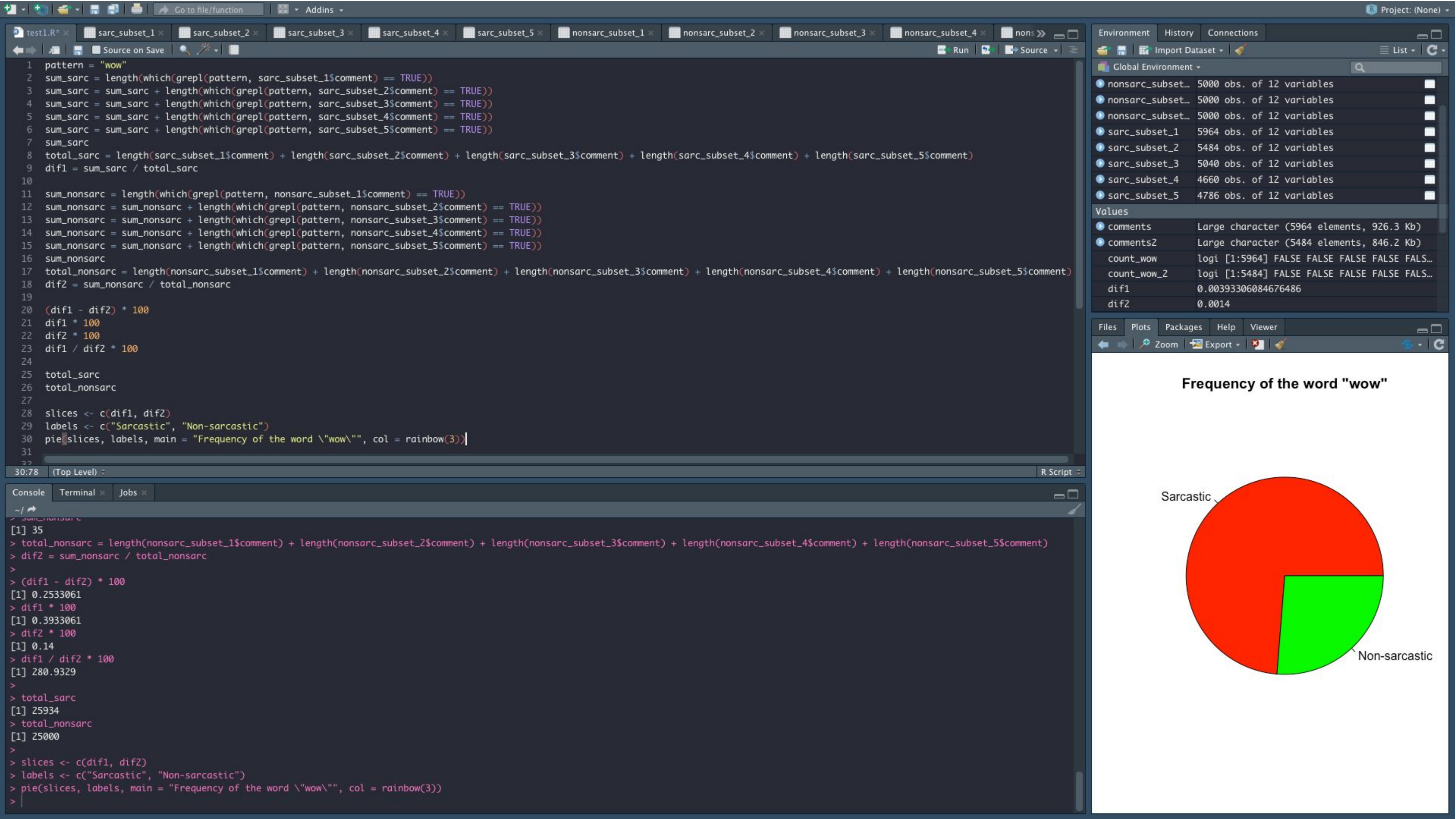
The problems with R results and solutions to them

Problems:

- High Margin of Error
- R “grepl(pattern, comments)” function

Solutions:

- Remove punctuation and special characters
- Add spaces in the beginning and the end of the string
- Use “ wow “ pattern instead of “wow”



```r
pattern = "wow"
sum_sarc = length(which(grepl(pattern, sarc_subset_1$comment) == TRUE))
sum_sarc = sum_sarc + length(which(grepl(pattern, sarc_subset_2$comment) == TRUE))
sum_sarc = sum_sarc + length(which(grepl(pattern, sarc_subset_3$comment) == TRUE))
sum_sarc = sum_sarc + length(which(grepl(pattern, sarc_subset_4$comment) == TRUE))
sum_sarc = sum_sarc + length(which(grepl(pattern, sarc_subset_5$comment) == TRUE))
sum_sarc
total_sarc = length(sarc_subset_1$comment) + length(sarc_subset_2$comment) + length(sarc_subset_3$comment) + length(sarc_subset_4$comment) + length(sarc_subset_5$comment)
dif1 = sum_sarc / total_sarc

sum_nonsarc = length(which(grepl(pattern, nonsarc_subset_1$comment) == TRUE))
sum_nonsarc = sum_nonsarc + length(which(grepl(pattern, nonsarc_subset_2$comment) == TRUE))
sum_nonsarc = sum_nonsarc + length(which(grepl(pattern, nonsarc_subset_3$comment) == TRUE))
sum_nonsarc = sum_nonsarc + length(which(grepl(pattern, nonsarc_subset_4$comment) == TRUE))
sum_nonsarc = sum_nonsarc + length(which(grepl(pattern, nonsarc_subset_5$comment) == TRUE))
sum_nonsarc
total_nonsarc = length(nonsarc_subset_1$comment) + length(nonsarc_subset_2$comment) + length(nonsarc_subset_3$comment) + length(nonsarc_subset_4$comment) + length(nonsarc_subset_5$comment)
dif2 = sum_nonsarc / total_nonsarc

(dif1 - dif2) * 100
dif1 * 100
dif2 * 100
dif1 / dif2 * 100

total_sarc
total_nonsarc

slices <- c(dif1, dif2)
labels <- c("Sarcastic", "Non-sarcastic")
pie(slices, labels, main = "Frequency of the word \"wow\"", col = rainbow(3))

```

```
nonsarc_subset…   5000 obs. of 12 variables
nonsarc_subset…   5000 obs. of 12 variables
nonsarc_subset…   5000 obs. of 12 variables
sarc_subset_1     5964 obs. of 12 variables
sarc_subset_2     5484 obs. of 12 variables
sarc_subset_3     5040 obs. of 12 variables
sarc_subset_4     4660 obs. of 12 variables
sarc_subset_5     4786 obs. of 12 variables
Values
comments          Large character (5964 elements, 926.3 Kb)
comments2         Large character (5484 elements, 846.2 Kb)
count_wow         logi [1:5964] FALSE FALSE FALSE FALSE FALS…
count_wow_2       logi [1:5484] FALSE FALSE FALSE FALSE FALS…
dif1              0.00393306084676486
dif2              0.0014
```

```
> sum_nonsarc
[1] 35
> total_nonsarc = length(nonsarc_subset_1$comment) + length(nonsarc_subset_2$comment) + length(nonsarc_subset_3$comment) + length(nonsarc_subset_4$comment) + length(nonsarc_subset_5$comment)
> dif2 = sum_nonsarc / total_nonsarc
>
> (dif1 - dif2) * 100
[1] 0.2533061
> dif1 * 100
[1] 0.3933061
> dif2 * 100
[1] 0.14
> dif1 / dif2 * 100
[1] 280.9329
>
> total_sarc
[1] 25934
> total_nonsarc
[1] 25000
>
> slices <- c(dif1, dif2)
> labels <- c("Sarcastic", "Non-sarcastic")
> pie(slices, labels, main = "Frequency of the word \"wow\"", col = rainbow(3))
>
```

MongoDB and Tableau

- MongoDB is a cross-platform document-oriented database program

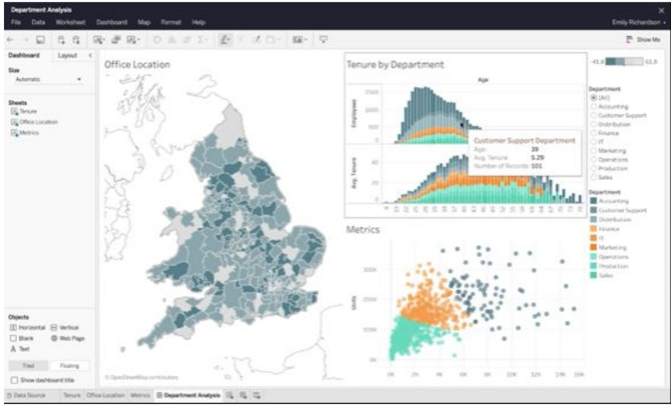
- MongoDB is classified as a NoSQL database program

- Relational databases store structured data like a phonebook. But for growing, unstructured data, a NoSQL database sets no limits, and allows you to add different types of data as your needs change.

- Tableau Software is an interactive data visualization software

What is Tableau?

- Tableau is a powerful and fastest growing data visualization tool.
- The best feature Tableau are
    - Data Blending
    - Real time analysis
    - Collaboration of data
- Tableau software doesn’t require any technical skills
- Popular visualizations:
    - Dashboards
    - Charts
    - Report Generation



Customer Support Department
Age: 39
Avg. Tenure 5.29
Number of Records 101

Problems with Using MongoDB and Tableau

Problems:

- Joining tables with the MongoBI Connector is a problem.
- Too many dependencies need to work & be installed to get MongoDB and Tableau working together.
- Not stable or reliable at times.
- Need enterprise version of MongoDB.

Where do we go from here?

Problems:

- Keep it simple and go with using .csv files.
- All though a NoSQL DB could give us quick speed, we really don’t need it in this case. We are using a subset of the whole data so putting everything into a database doesn’t seem useful for our dataset size. If we were to scale up our data that we want to use, then we could consider adding it into a NoSQL DB.
- Allows people to work independently from others to create their own metrics.

Results We Hope to Get

- At first, a statistical model that can identify a sarcastic comment with 70% accuracy.
  - The model will be based on the occurrence of things like key words, phrases, and the topic of conversation.

- If time permits we hope to develop a code body that is easily accessible.
  - We want to leave more than just a model to the NLP community.
  - And we want our code to be easy to use and to expand on, for those who desire.

Conclusion

- We would benefit greatly from more domain knowledge.
- python, R, pycharm, and R-studio.
- We have a clean subset of the original data.
- We’ll each run our own metrics.
- We’ll collectively analyze the results and collectively add to the model.

The end.
This has been, the best presentation ever.

Project Source Code:
https://github.com/steve3p0/cs510ds

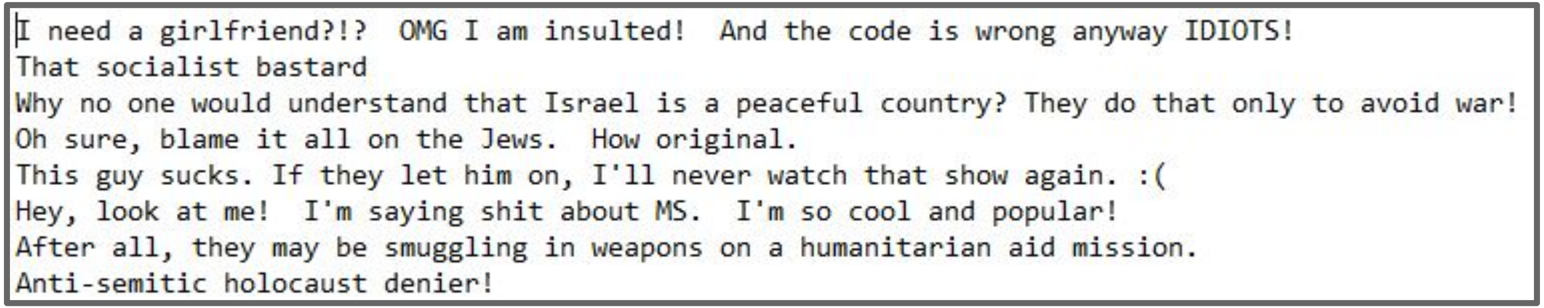
Questions?

Appendix

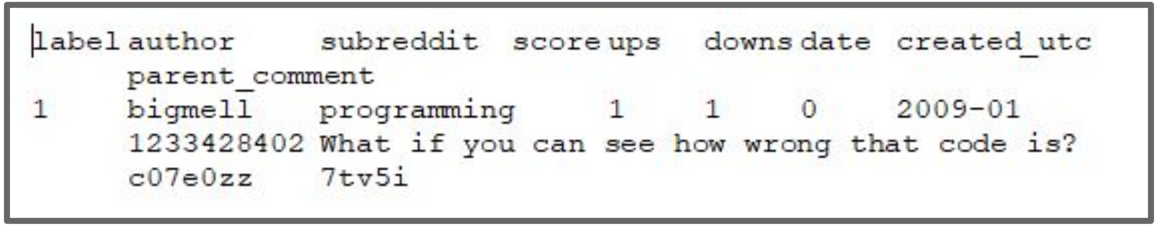
The Data: Our Structure

- Each type of data, sarcastic or not, is stored across 2 files.

- A file for the posts, a file for their metadata.



```
I need a girlfriend?!?  OMG I am insulted!  And the code is wrong anyway IDIOTS!
That socialist bastard
Why no one would understand that Israel is a peaceful country? They do that only to avoid war!
Oh sure, blame it all on the Jews.  How original.
This guy sucks. If they let him on, I'll never watch that show again. :(
Hey, look at me!  I'm saying shit about MS.  I'm so cool and popular!
After all, they may be smuggling in weapons on a humanitarian aid mission.
Anti-semitic holocaust denier!
```



```
label author     subreddit  score ups   downs date  created_utc
      parent_comment
1     bigmell    programming     1     1     0     2009-01
      1233428402 What if you can see how wrong that code is?
      c07e0zz    7tv5i
```