Hello!

Members:

Willie Xia - xiaw@rpi.edu

Iris Liu - liui2@rpi.edu

Min Yue - yuem@rpi.edu

Helen Yuan - yuanh2@rpi.edu

Our Updates is Included in the Latter Portion of our Presentation

Utilizing Deep Learning To Classify Tweets

TextBlob

NLTK (**N**atural **L**anguage **T**ool**K**it)

scikit-learn

spaCy

pytext

VADER (**V**alence **A**ware **D**ictionary and s**E**ntiment **R**easoner)

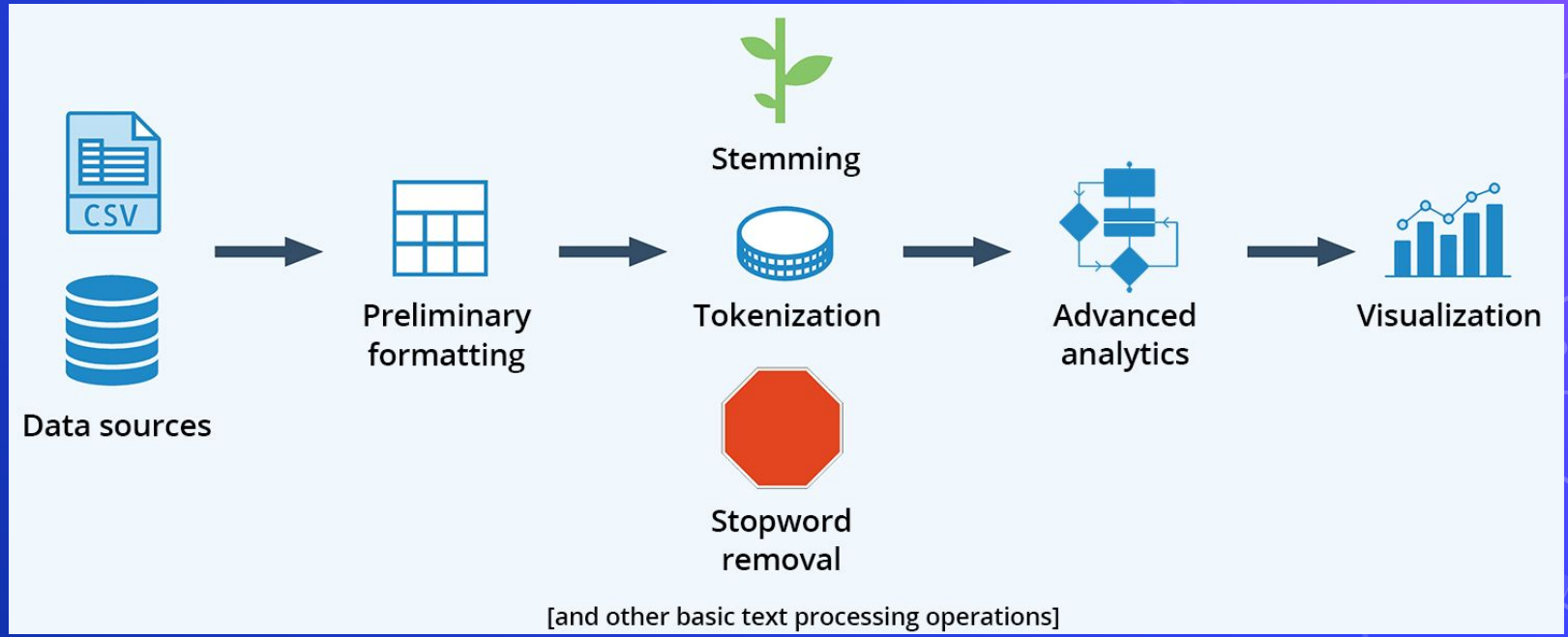
Rule-based

Automatic

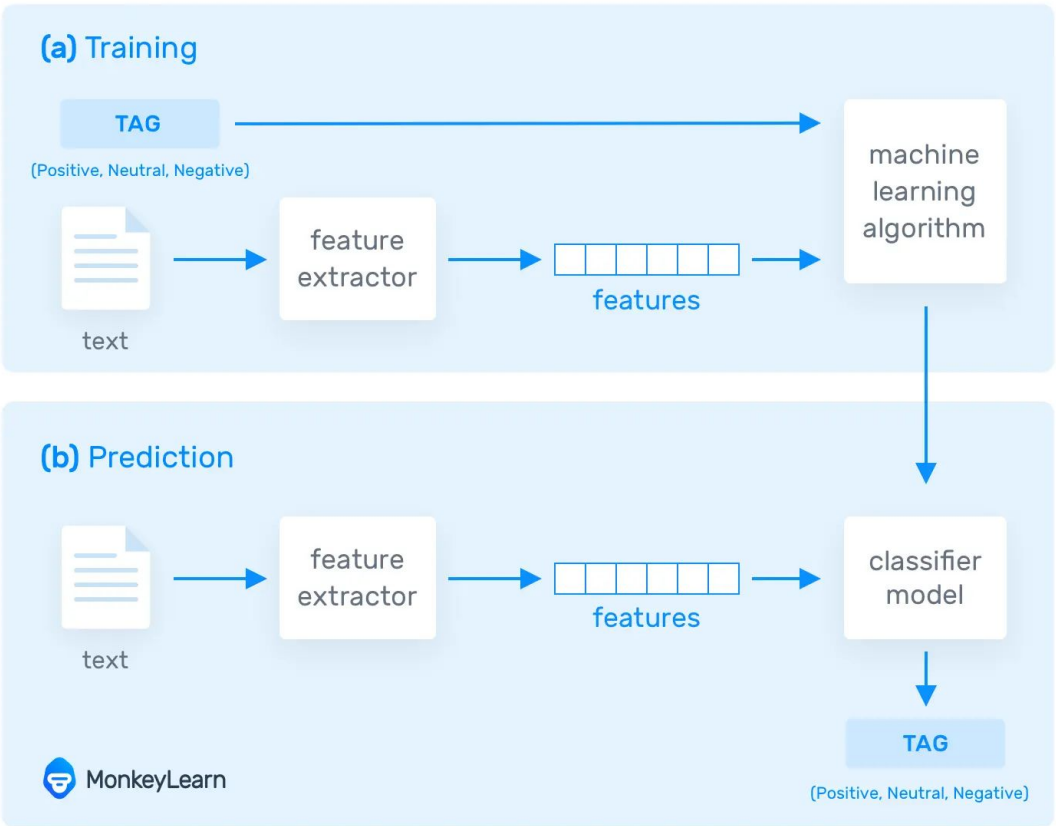
Hybrid

The Approach...

A Pseudo-Text Blob Network



https://www.softwareadvice.com/resources/what-is-text-analytics/



https://monkeylearn.com/sentiment-analysis/

Naive Bayes Classification

The Advantages...

- **Flexible** and can work around multiple parameters
- Easily **modifiable** to calculate different properties
- **Intuitive** and not as “black-box” as other sentiment models

The Disadvantage...

- Is **not in depth** with one property
- **Not as accurate** as other pre-built sentiment models
- **Vulnerable to vocabulary** that the model wasn't trained with

How our Naive Bayes Model works:

Tweet 1: “This steak is great.” → Tokens: [THIS, STEAK, IS, GREAT]

Predetermined Sentiment: 1 (positive)

Our Model: ↓

“THIS”	1
“STEAK”	1
“IS”	1
“GREAT”	1

How our Naive Bayes Model works:

Tweet 2: “this steak is bad” → Tokens: [THIS, STEAK, IS, BAD]

Predetermined Sentiment: -1 (negative)

Our Model: ↓

“THIS”	1, -1
“STEAK”	1, -1
“IS”	1, -1
“GREAT”	1
“BAD”	-1

How our Naive Bayes Model works:

Result:

Our Model:

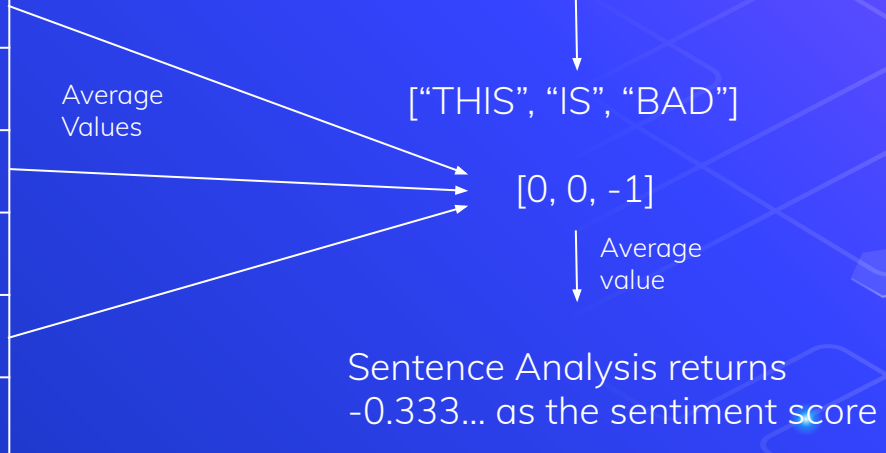
“THIS”	1, -1	Average value: 0 (neutral)
“STEAK”	1, -1	Average value: 0 (neutral)
“IS”	1, -1	Average value: 0 (neutral)
“GREAT”	1	Average value: 1 (positive)
“BAD”	-1	Average value: -1 (negative)

How our Naive Bayes Model works:

Result:

Our Model:

“THIS”	1, -1
“STEAK”	1, -1
“IS”	1, -1
“GREAT”	1
“BAD”	-1



How our Naive Bayes Model works:

“I don’t like” → Tokens: [I, DONT, LIKE]

Predetermined Sentiment: -1 (negative)

“I”	“DONT”	-1
	“LIKE”	-1
		-1
“DONT”	“I”	-1
	“LIKE”	-1
		-1
“LIKE”	“I”	-1
	“DONT”	-1
		-1

How our Naive Bayes Model works:

“I like” ⟶ Tokens: [I, LIKE]

Predetermined Sentiment: 1 (positive)

“I”	“DONT”	-1
	“LIKE”	-1, 1
		-1, 1
“DONT”	“I”	-1
	“LIKE”	-1
		-1
“LIKE”	“I”	-1, 1
	“DONT”	-1
		-1, 1

How our Naive Bayes Model works:

“I”, “DONT”	average value: -1
“I”, “LIKE”	average value: 0
“I”	average value: 0
“DONT”, “I”	average value -1
“DONT”, “LIKE”	average value -1
“DONT”	average value -1
“LIKE”, “I”	average value 0
“LIKE”, “DONT”	average value -1
“LIKE”	average value 0

“I”	“DONT”	-1
	“LIKE”	-1, 1
		-1, 1
“DONT”	“I”	-1
	“LIKE”	-1
		-1
“LIKE”	“I”	-1, 1
	“DONT”	-1
		-1, 1

How our Naive Bayes Model works:

The performance of our model's sentiment analysis all depends on the data we train it with.

- Accuracy
- Balance (bias)
- Amount
- Optional: multiple sources?

Current state of Naive Bayes Sentiment Analysis

- More/Deeper Layers
- Weighting, and Smoothing
- Word definitions
- True understanding of words/topics

So far...

Filtration using Panda framework

Naive Bayes model

The Sentiment Scale

Sample Output 1 (stay home, and stay safe!)

STAY

STAY: 0.32551282051282066

STAY, AND: 0.3387962962962965

STAY, HOME: 0.4065000000000002

STAY, SAFE: 0.5

Average value: 0.3927022792022793

Sample Output 1 (stay home, and stay safe!)

AND

AND: 0.14948988511488512

AND, STAY: 0.3387962962962965

AND, HOME: 0.3048648648648651

Average value: 0.2643368209201557

Sample Output 1 (stay home, and stay safe!)

SAFE

SAFE: 0.5

SAFE, STAY: 0.5

SAFE, HOME: 0.5

Average value: 0.5

sentence sentiment:
0.381602887263304

Sample Output 2 (coronavirus is so bad and I hate it)

CORONAVIRUS

CORONAVIRUS: 0.15625

CORONAVIRUS, AND: 0.25

CORONAVIRUS, I: 0.25

Average value: 0.21875

Sample Output 2 (coronavirus is so bad and I hate it)

IS

IS: 0.08925246512746521

IS, AND: 0.28905935613682093

IS, IT: -0.008189655172413792

IS, I: 0.36215277777777777

IS, SO: 0.36

Average value: 0.21845498877393005

Sample Output 2 (coronavirus is so bad and I hate it)

BAD

BAD: -0.41025641025641024

BAD, I: -0.1666666666666666

BAD, AND: -0.5374999999999999

BAD, SO: -0.6999999999999998

Average value: -0.4536057692307691

sentence sentiment:
-0.020170796696350 11

“

“Weather is so gorgeous but we still have Corona out here…”

“

“Weather is so gorgeous but we still have Corona out here…”

TextBlob sentence sentiment:

0.7

Our sentence sentiment:

0.3802296180222242

Issue: Dataset Problems

- Sentiment ratings in the training data is **skewed** positive
- Some Tweets **appear as neutral** when they should be negative or positive
- Examples:
  - “Kill it with fire” - **0.0**
  - “if corona gets sean payton ima k*** myself” - **0.0**

Next Step

Achieve by Aug 21st

- Graph representation
- Using recent tweets to test our AI
- Feature: Tweet prediction (month, who, topics)

Further Goals

- Build our own training datasets
- Confidence level of positive, negative, and neutral posts
- Input a topic and will show the sentiment of the topic from Twitter
- Add GUI
- Remove stopwords/Learn word weight
- Add deeper layers to the model

Thanks!

Any questions?

References

- https://github.com/sloria/TextBlob
- https://www.softwareadvice.com/resources/what-is-text-analytics/
- https://monkeylearn.com/sentiment-analysis/

UPDATES

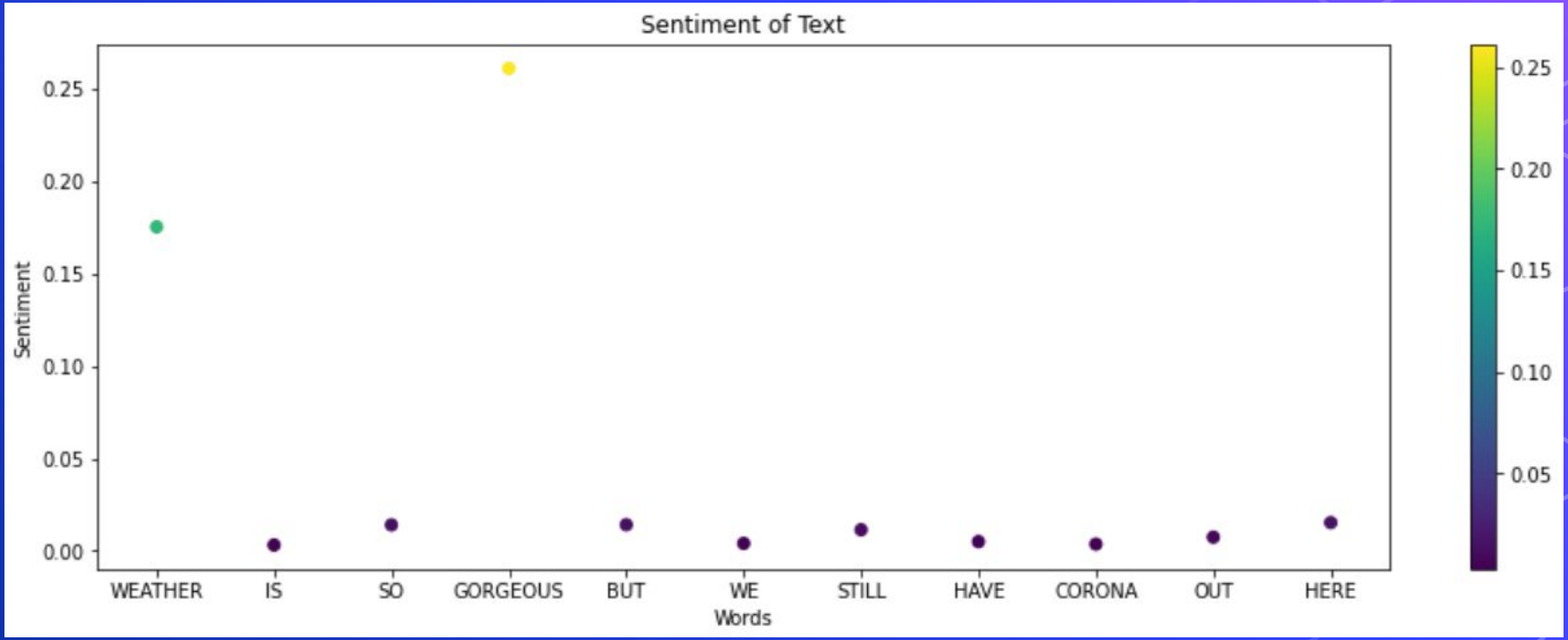
Changes Made Since 8/17

- Implemented word weight distribution in order to achieve more accurate values, as some words may have higher sentimental impact to the sentence than others.
- Created plots demonstrating our model and its output.
- Included substantially more data to train and test our model.
- Was able to execute more testing with more data.
- Added a tag system, where words could be related to tags given in training

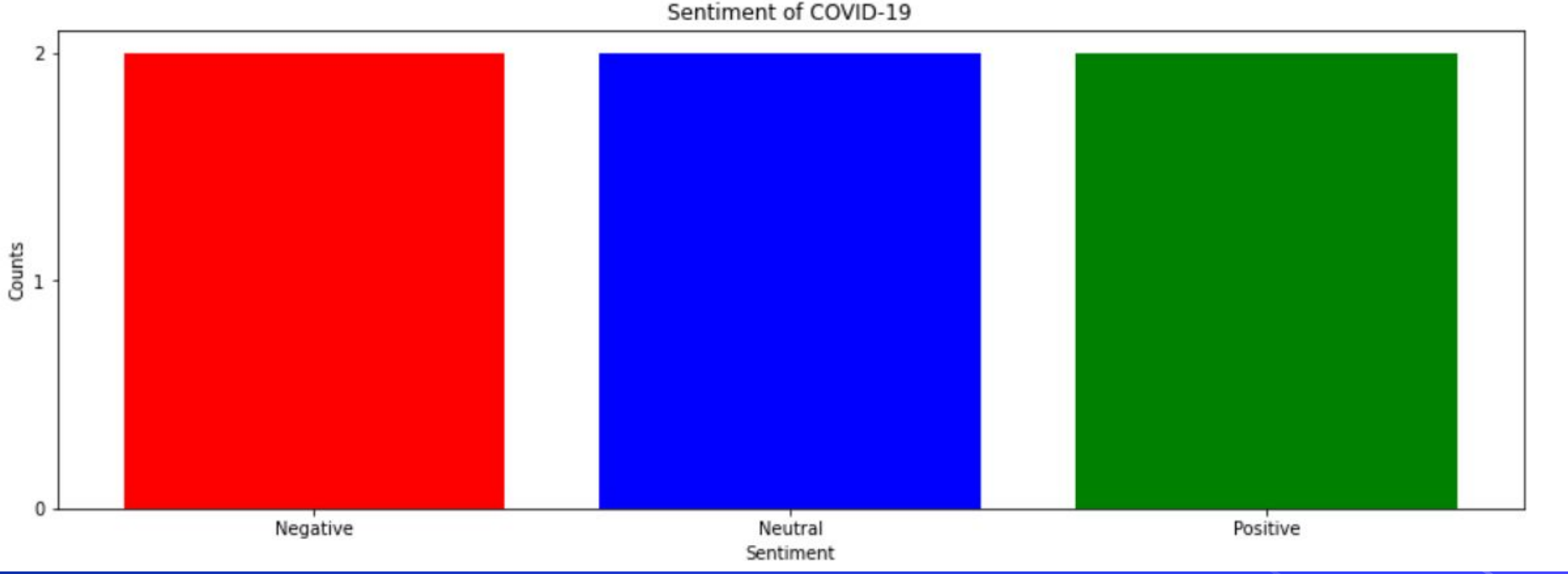
Results

- Achieved a fully functional model that determines the sentiment value of a sentence; it accounts for word weight and previously encountered word pairings.
- Included a tag predictor to our model, which estimates what tag a sentence belongs to based off of what tags were given during training.

Data Visualization



Data Visualization



Comparisons

Tweet	**TextBlob**	**NLTK**	**Google Cloud Natural Language API**	**Public Emotions**
Weather is so gorgeous but we still have Corona out here…	0.7	0.4693	0.3	0.513980579743914
corona is destroying everything and I'm so sad	-0.483333333	-0.822	-0.8	-0.0972917887745161
Kill it with fire	0	-0.7964	-0.5	0.110075443266489

Responsibilities

- Willie Xia
  - Planned and initialized the training method - focused mostly on parsing and tokenization. Managed group work spaces.
- Iris Liu
  - Crafted the datasets and helped out through various small tasks like comparing our group's result to other existing sentiment analyzers
- Min Yue
  - Planned and wrote the training method, the analysis method, and the tag system for the Naive Bayes model
- Helen Yuan
  - Crafted the datasets, created data visualizations, manage GitHub repo, helped out with the analysis method, and worked on documented our work

Future Goals

- Deeper layers
- Better of understanding english
- Nouns? Adjectives? Etc.
- Big one: Word definitions
- Sentence formulation