Too Soon:

Classifying Questionable Headlines

JEFF WARCHALL

Sorting Comedic Headlines

Can we use NLP to tell the difference between:

The Best of The Onion Subreddit - https://www.reddit.com/r/TheOnion/

- 15,053 of the most recent posts
- There are only ~20,000 posts total

Sadly, this is not The Onion Subreddit - https://www.reddit.com/r/nottheonion/

- 15,099 of the most recent posts
- Nearly 500,000 total posts

Which Are Real:

DC Congress Woman Bizarrely Denies Letting Zebras Loose

Rapper has Gold Chains Surgically Implanted into his Head

Bishop Quits Church after Falling in Love with Satanic Erotica Writer

Seriously, This Time:

Sarcasm

The Onion:

Benefits of Open Office Not Extended to CEO

Not the Onion:

Delta CEO Reveals He's Still Refusing to Call it the Delta Variant

Seriously, This Time:

Word Frequency

The Onion:

Man's Life Riddled with Continuity Errors

Not the Onion:

Biden Administration Worried **Taliban** isn't Diverse Enough.

Seriously, This Time:

Non Sequitur

The Onion:

Activists Petition **Cupcake Kingdom** to Address **Affordable Housing** Crisis

Not the Onion:

Millennials trying to Live out of Storage Units in Recent Housing Shortage

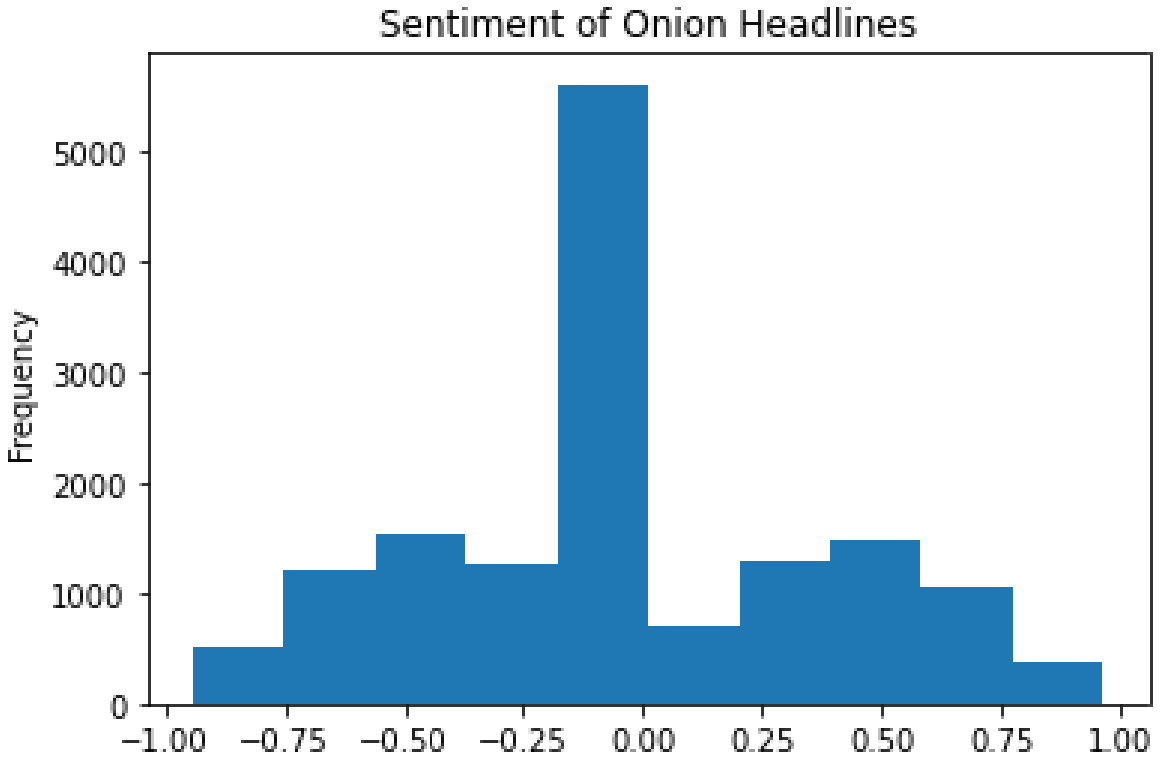
Possible Strategies

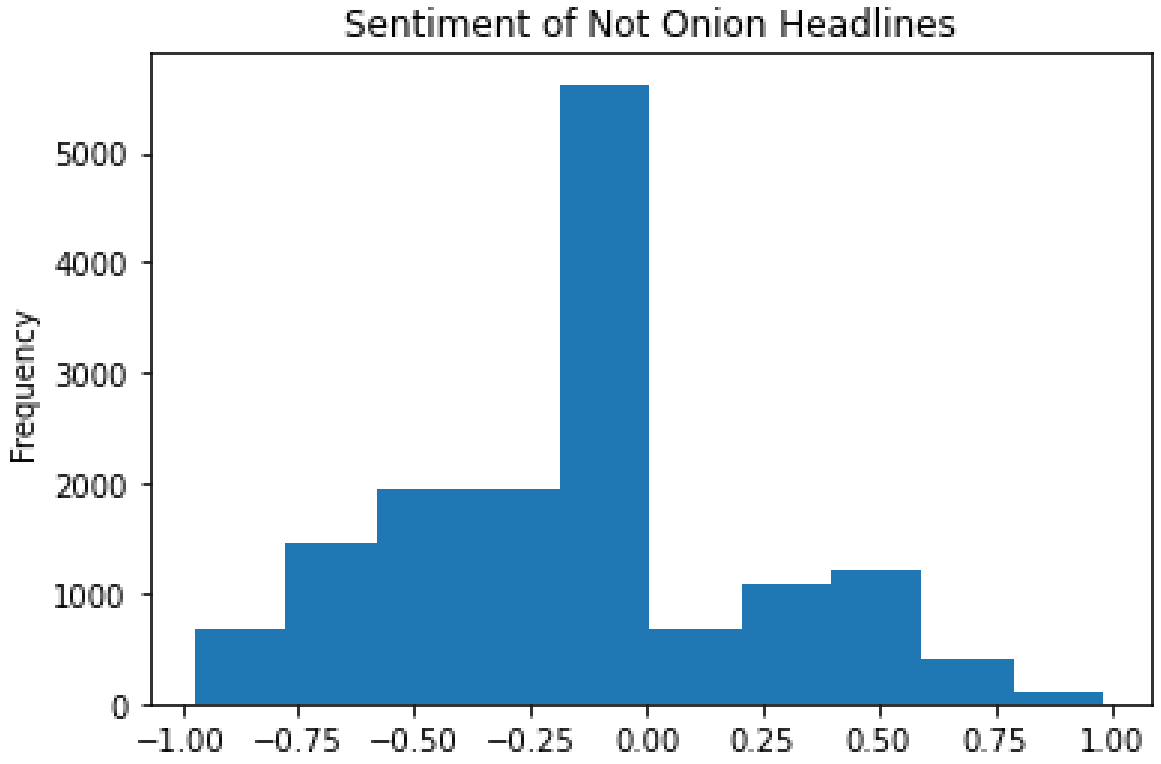
Sarcasm: Sentiment Analysis

Word Frequency: Term Frequency – Inverse Document Frequency

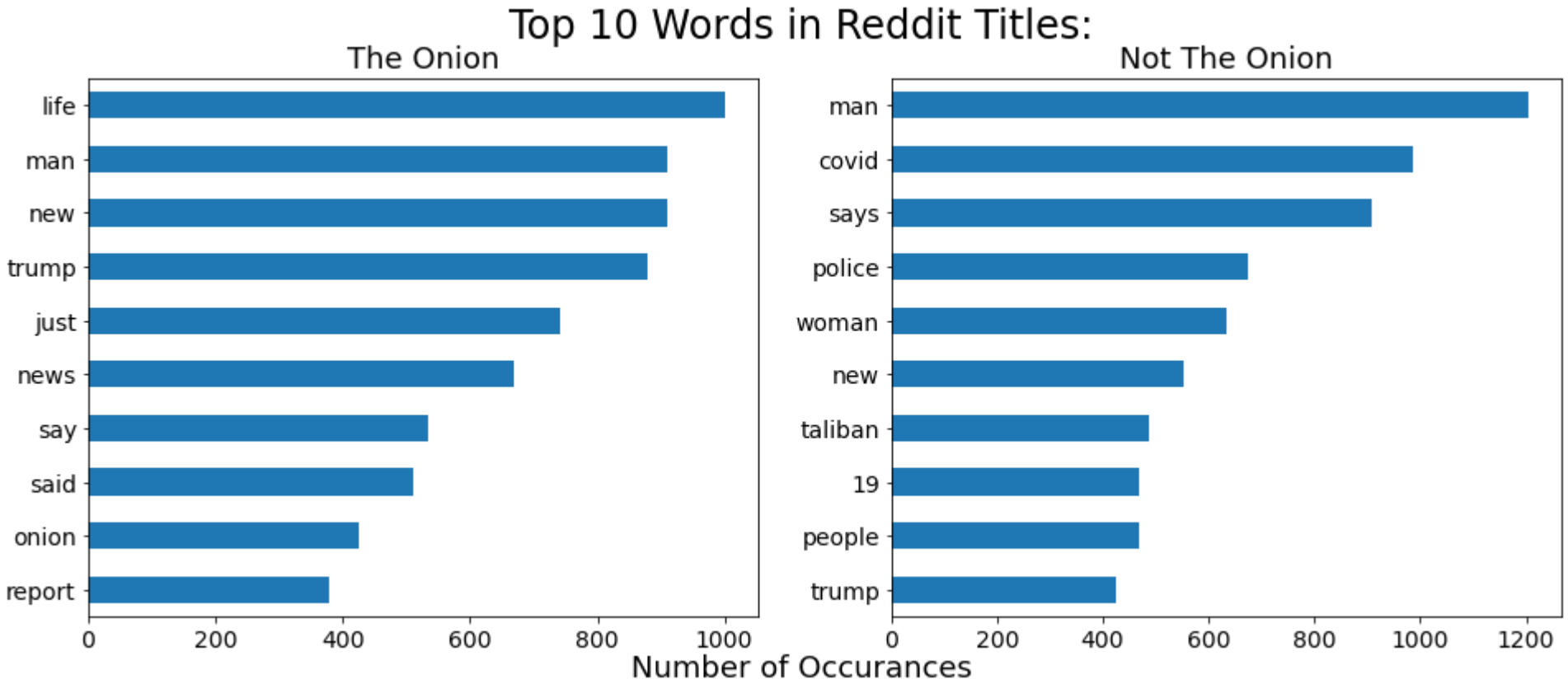
Non Sequitur: Word Vectorization

Sentiment Analysis





Word Frequency



Non Sequitur

```
[42]: final_word = glove.get_vector('side')
      for word in "salad clearly hamburger toppings".split():
          new_word = glove.get_vector(word)
          final_word = final_word + new_word

      glove.similar_by_vector(final_word)

[42]: [('cheese', 0.786815345287323),
       ('sandwiches', 0.7859990000724792),
       ('dressing', 0.7758222818374634),
       ('chicken', 0.7721953392028809),
       ('pasta', 0.7695923447608948),
       ('salad', 0.7670999765396118),
       ('sandwich', 0.7597503066062927),
       ('soup', 0.7590339779853821),
       ('cooked', 0.7521399259567261),
       ('sausage', 0.7491800785064697)]
```

```
[65]: from IPython.display import clear_output

[69]: total = len(onion_df.columns)
      iters = 0
      for word in onion_df.columns:
          clear_output()
          iters += 1
          print(f"{iters} of {total} completed.")
          try:
              onion_df[word] = glove.get_vector(word)
          except:
              onion_df[word] = np.nan

47 of 19761 completed.

ValueError                                Traceback (most recent call last)
```

Unfortunately, this would have taken 22 hours to run

https://www.technologyreview.com/2015/09/17/166211/king-man-woman-queen-the-marvelous-mathematics-of-computational-linguistics/

Support Vector Machine (SVM)

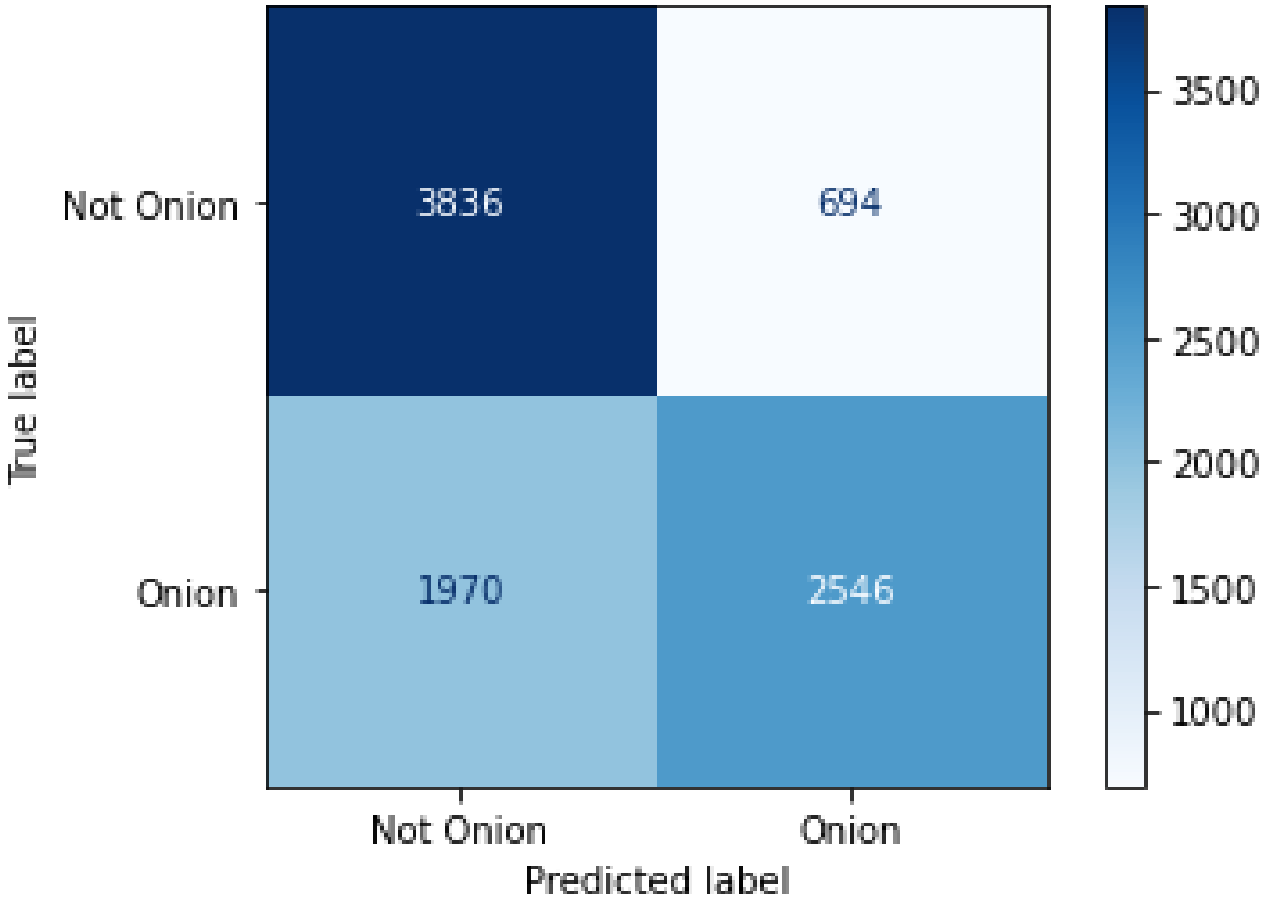
SVM

Accuracy: 70.6%

Recall: 56.4%

Precision: 78.6%

F_1 Score: 65.7%



Logistic Regression

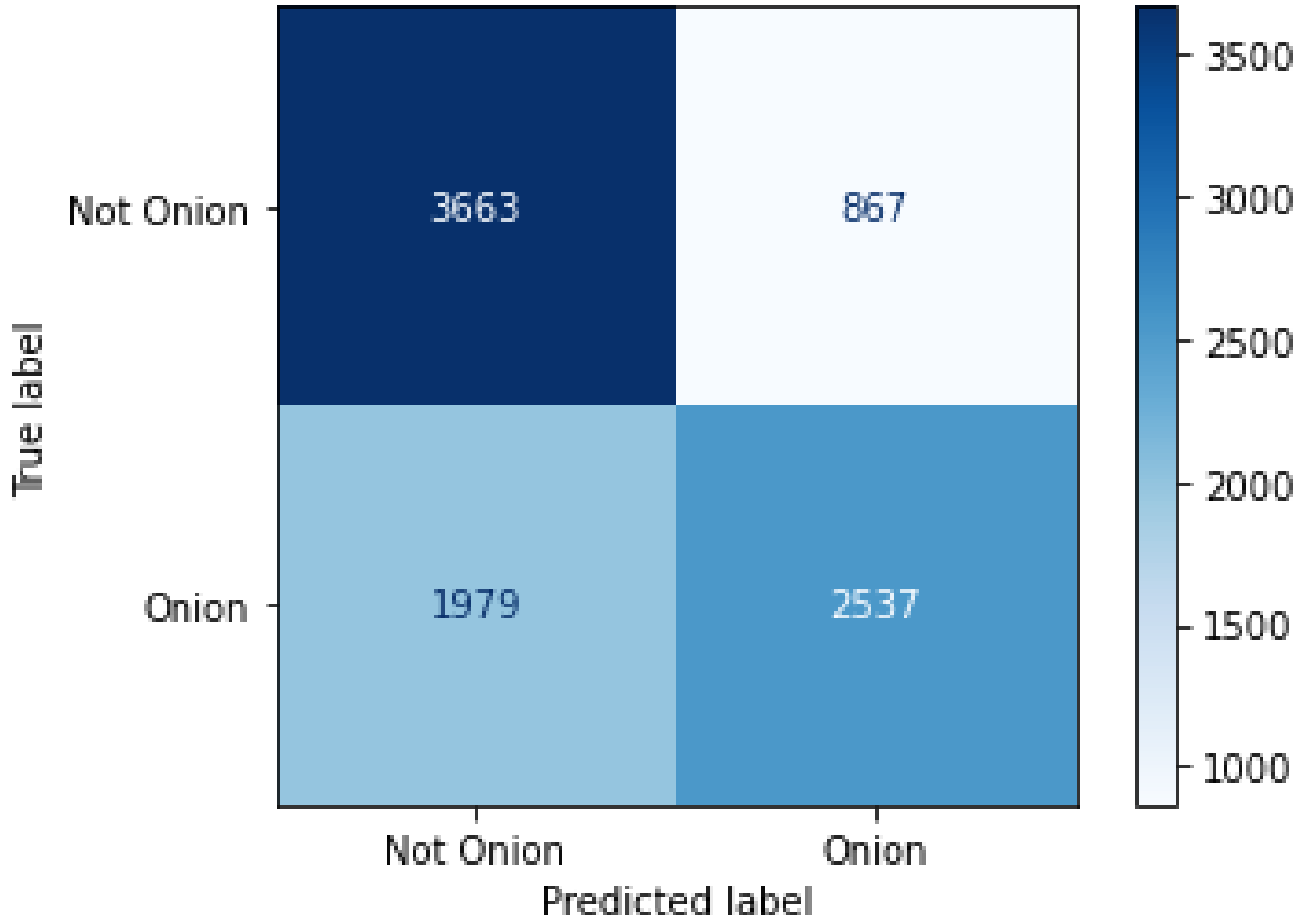
Logistic Regression

Accuracy: 68.5%

Recall: 56.2%

Precision: 74.5%

F_1 Score: 64.1%



Random Forest

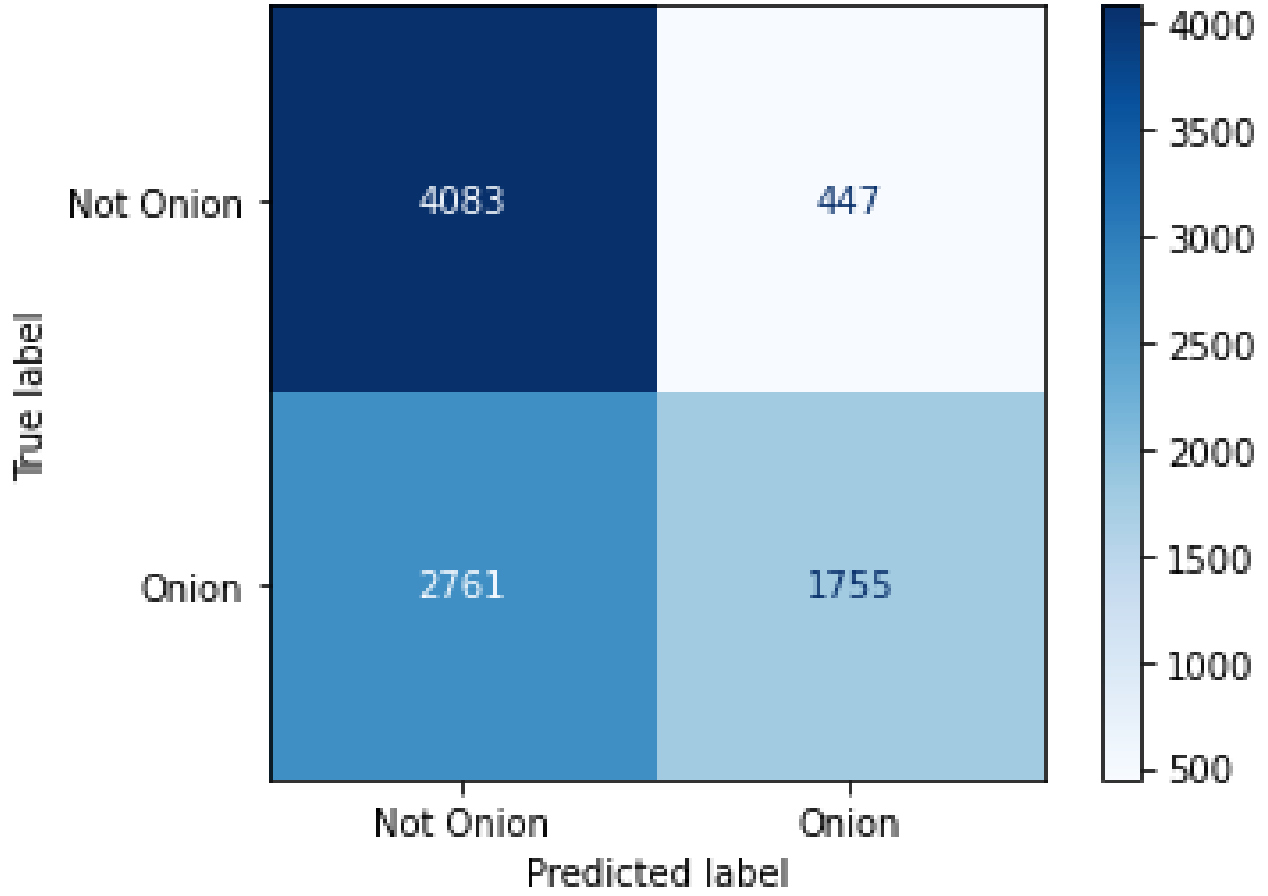
Random Forest

Accuracy: 64.6%

Recall: 38.9%

Precision: 79.9%

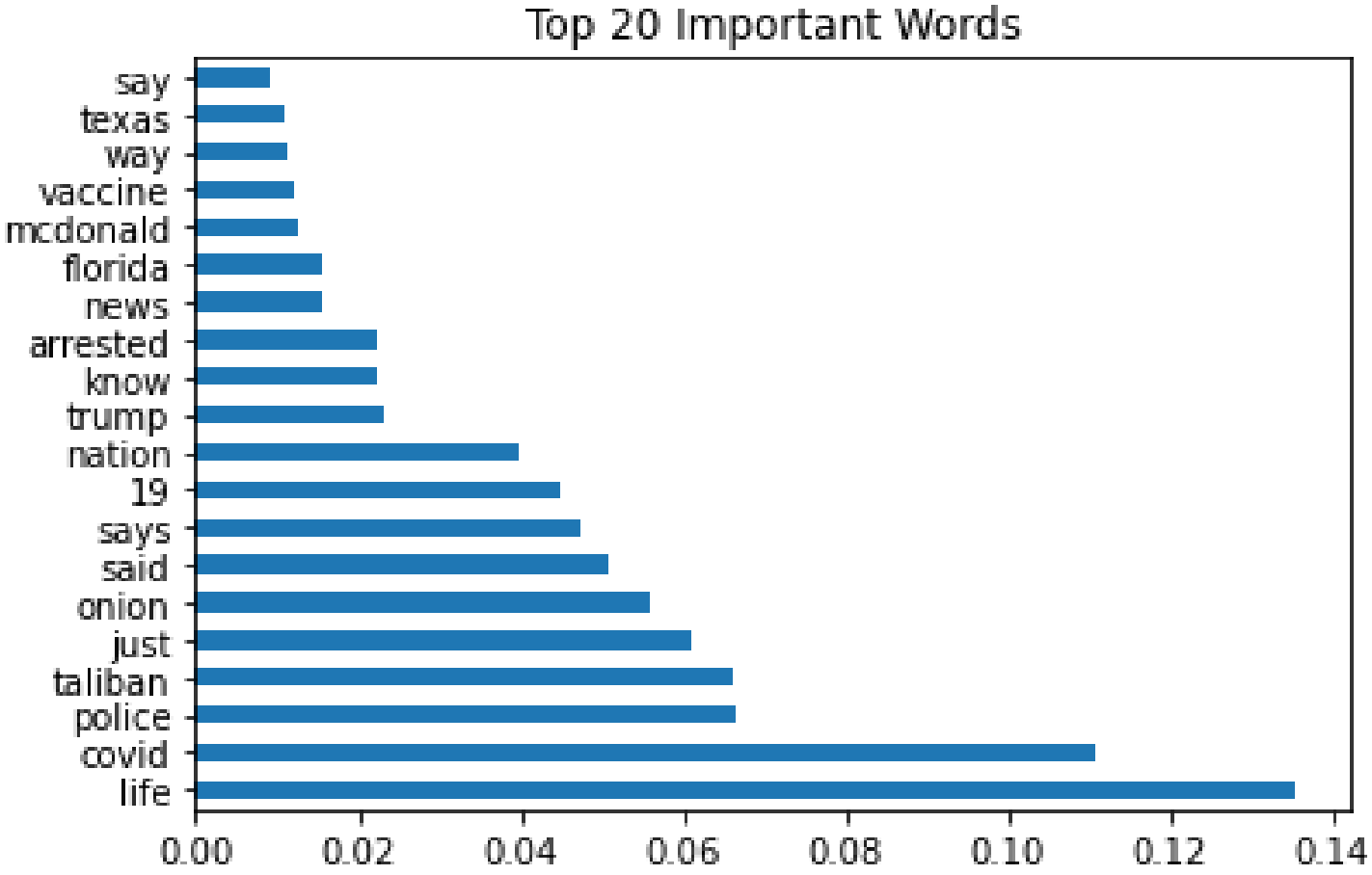
F_1 Score: 52.2%



Ensemble

SVM	Logistic Regression	Random Forest	Ensemble
			Train Accuracy: 69.1%
Accuracy: 70.6%	Accuracy: 68.5%	Accuracy: 64.6%	Accuracy: 69.2%
Recall: 56.4%	Recall: 56.2%	Recall: 38.9%	Recall: 53.7%
Precision: 78.6%	Precision: 74.5%	Precision: 79.9%	Precision: 77.7%
F_1 Score: 65.7%	F_1 Score: 64.1%	F_1 Score: 52.2%	F_1 Score: 63.5%

Feature Importance



Conclusion

The SVM model performed better than the rest of the models in all categories except for the precision of the Random Forest.

Ensemble model has very low variance

Too Soon - It seems like the best way to split these data was on keywords that are:

- Current
- Tragic

Limitations and Future Work

Time is likely a *very* important factor:

- 15,000th Onion Article was about Jeb Bush and Donald Trump
- 15,000th Non Onion Article was about Jeff Bezos going to space

Computer resource issues (N-grams, Word Vectorization)

Can humans do better than 70%

Thank You!

ANY QUESTIONS?