Fake News Classification

An Introductory Text Analysis

David Curry
November 14, 2017

Python Notebook Link

Analysis Motivation

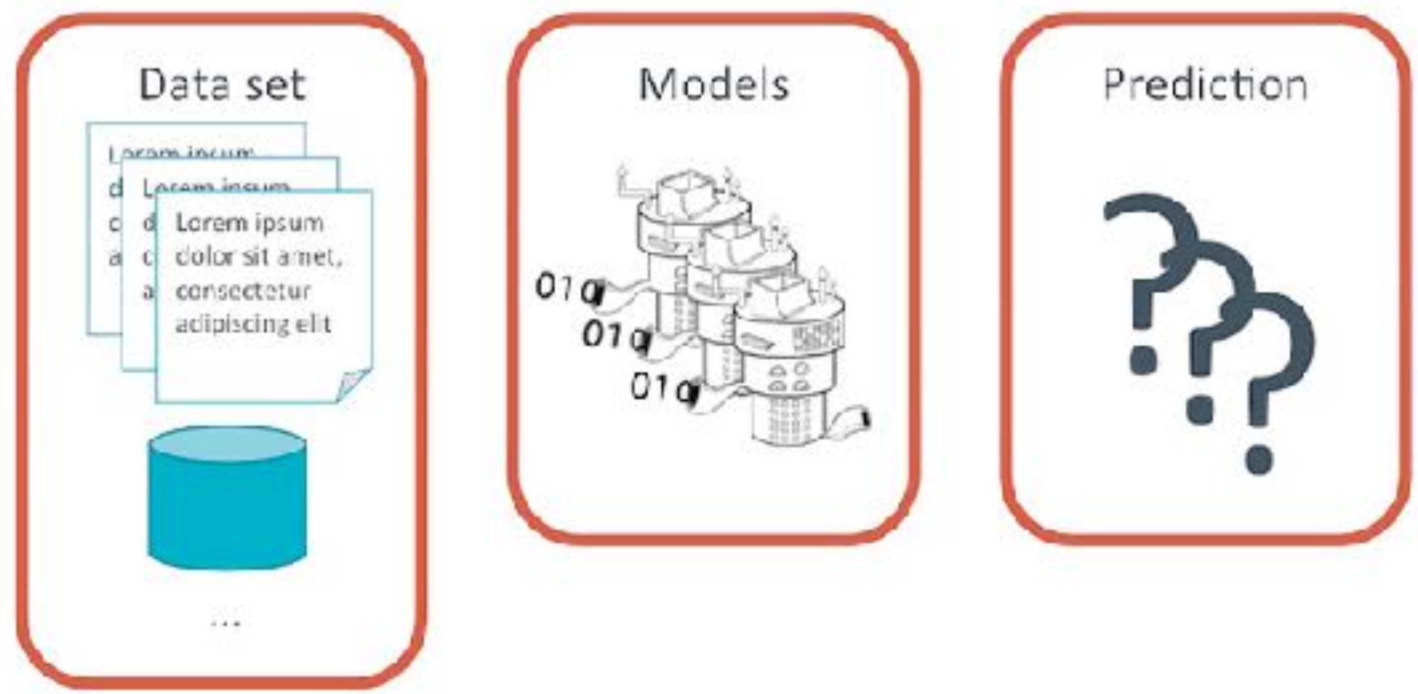
- Fake News has become ubiquitous, even becoming **Word of the Year** according to Collins Dictionary. The moral and legal implications are huge

- **Goal:** Given a random news article from an unknown source predict whether it is fake or real

- **Binary Classification**
  Simplest Model First: Naive Bayes

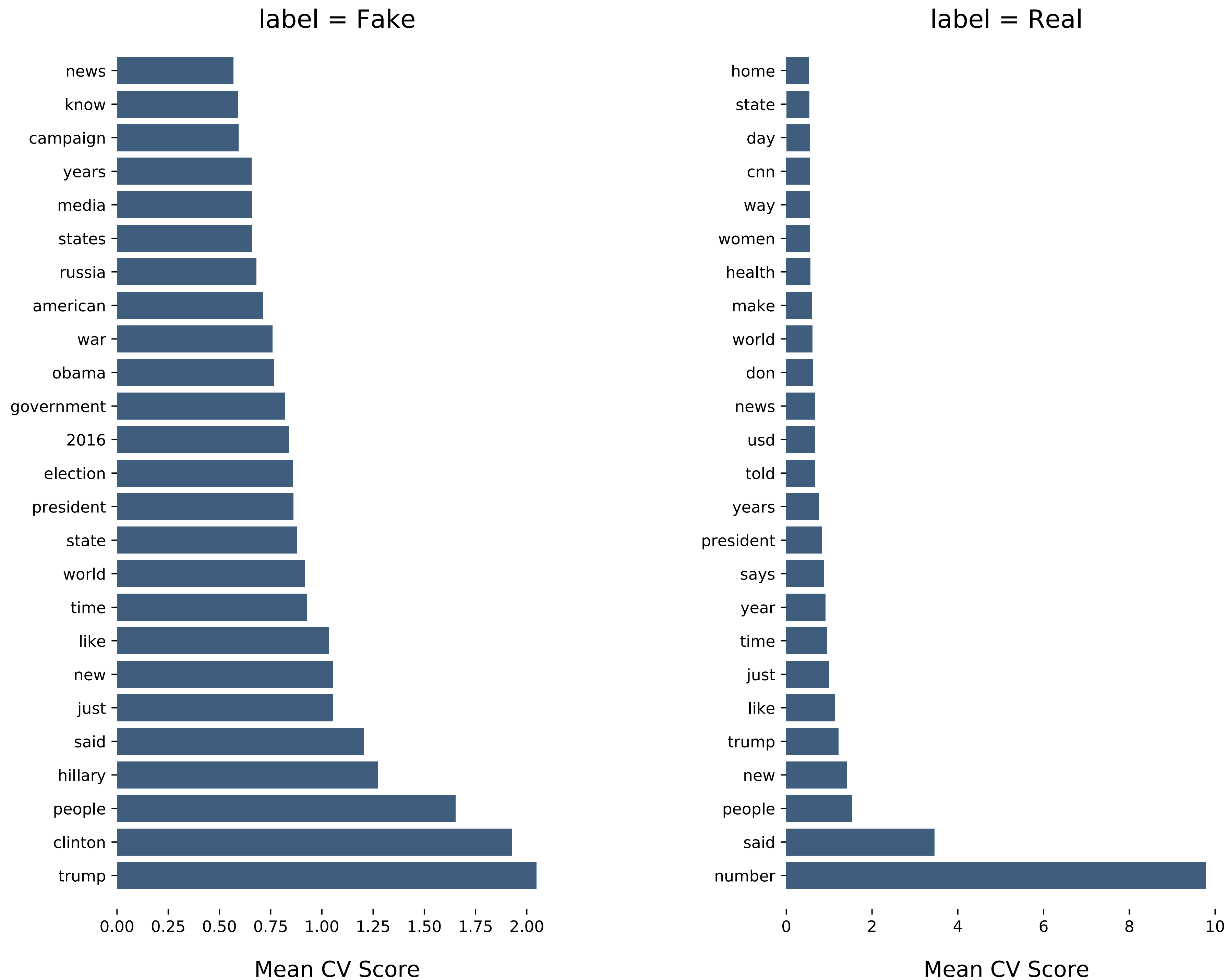
- **Training Features**
  Bag of Words Method: Word occurrences are the features

- **Inherent Bias in the Target Classes**
  - What is fake or real depends on who is building the model
  - Training for maliciousness, not "truth"
  - Writing style, intended audience are more correlated to target classes than objective truth



First Look at Data

- **Fake News**
  - Kaggle Dataset contains 12,500 news articles(54M)
  - Sources: Brietbart, InfoWars, DrudgeReport, misc. malicious "news"
- **Reals News**
  - Articles scraped from 8 different news sites NYT, CNN, FOX, AllSides, etc.
  - 2,800 articles(11M) and growing
  - Jupyter Notebook Link



Top 25 Average Word Occurrences Per Article

Features and Model

- **Word Occurrences as Features**
  - Standard Out-of-the-Box(OOB) Method
  - Each row is an article. Each column is a unique word in the entire corpus
  - Column entries are word occurrences(See below for toy example

docs = ["You can catch more flies with honey than you can with vinegar.",
"You can lead a horse to water, but you can't make him drink."]

but	can	catch	drink	flies	him	honey	horse	lead	make	more	than	to	vinegar	water	with	you
0	0	2	1	0	1	0	1	0	0	0	1	1	0	1	0	2
1	1	2	0	1	0	1	0	1	1	1	0	0	1	0	1	0

- **Additional Features: n-grams**
  - Takes into account relationships amongst words
  - Uni-gram is a single word(our first model is a 1-gram model)
  - n-grams looks n words in front and back of a word

- **Model: Naive Bayes Classifier**
  - Very popular due to good performance and ease of interpretation
  - Builds a probability table for each class and each unique word to be found in it
  - Sci-Kit's Multinomial Bayes Classifier: LINK

Initial Results

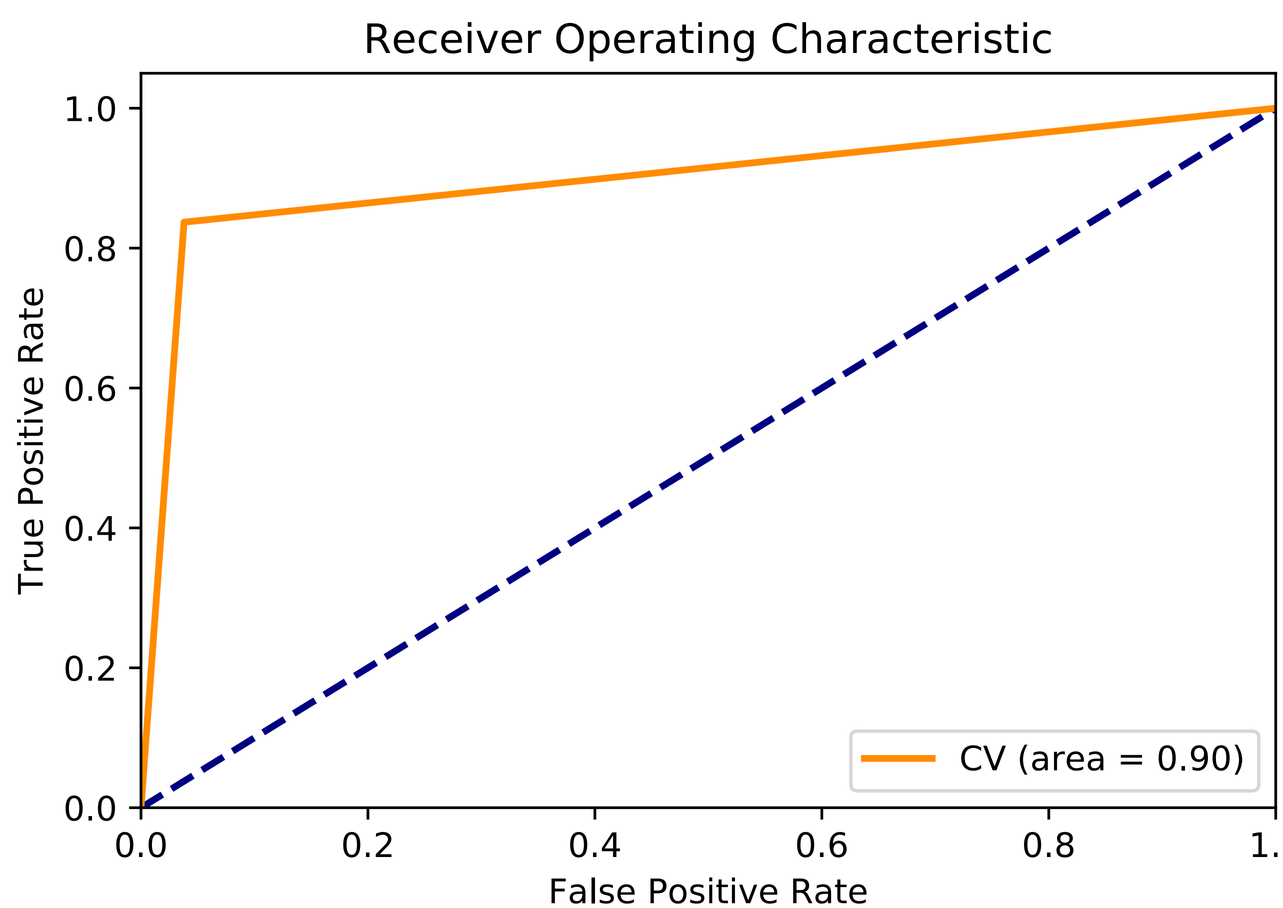
Fake = Positive

Confusion Matrix: 1-gram	
11897	460
470	2385

* Simple model(1-gram) out-of-the-box has 93% accuracy
* Metric of choice depends on our needs:
    * Flag the highest % of fake news submitted: **High Sensitivity(TPR)**
    * Avoid unfairly flagging real news: **Low FPR(1-specificity)**
    * Purity of flagged fake news: **High Precision**

* **OOB(out of the box) performance is reasonable**

```
accuracy    : 0.938
specificity : 0.835
sensitivity : 0.962
precision   : 0.962
f1score     : 0.962
```



Optimization

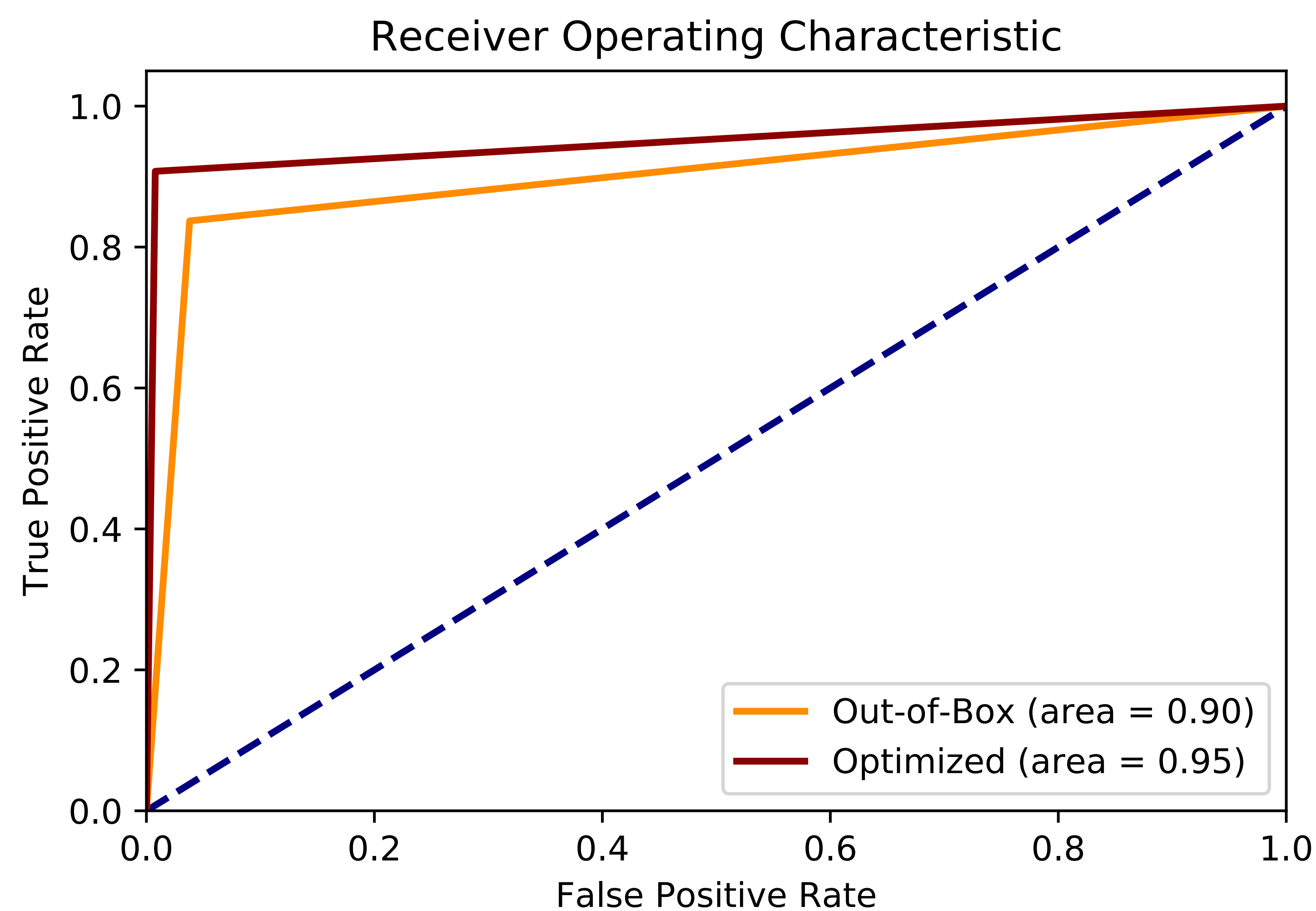
Fake = Positive

Confusion Matrix: 5-grams	
12261	96
264	2591

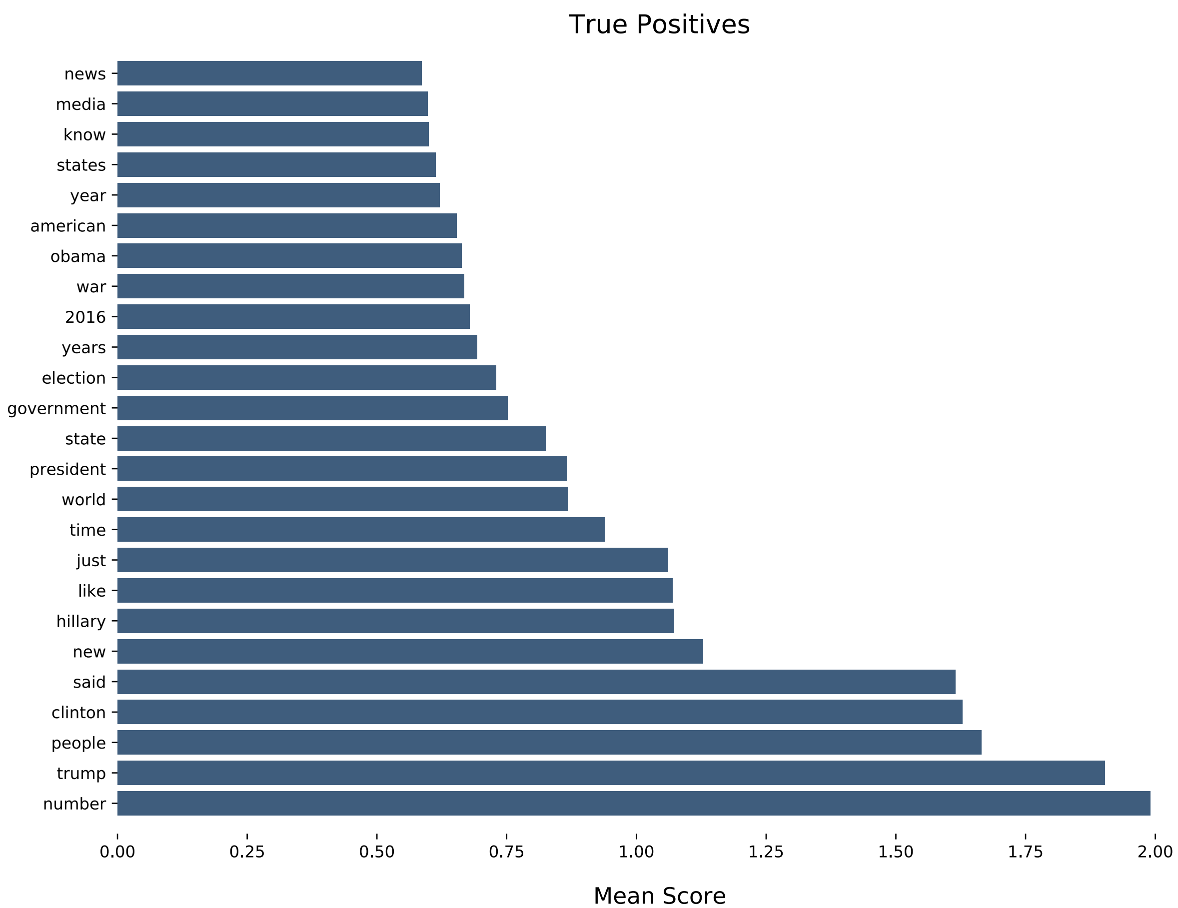
- Tuning the Hyperparameters:
  - n-grams = [0, 1, 2, 3, 4, 5]
  - stop_words(True of False)
  - Smoothing Parameter = (1, 0, 1e-1, 1e-2, 1e-3, 1e-4),
    - Used to account for words in the test set that might not have been in the training set

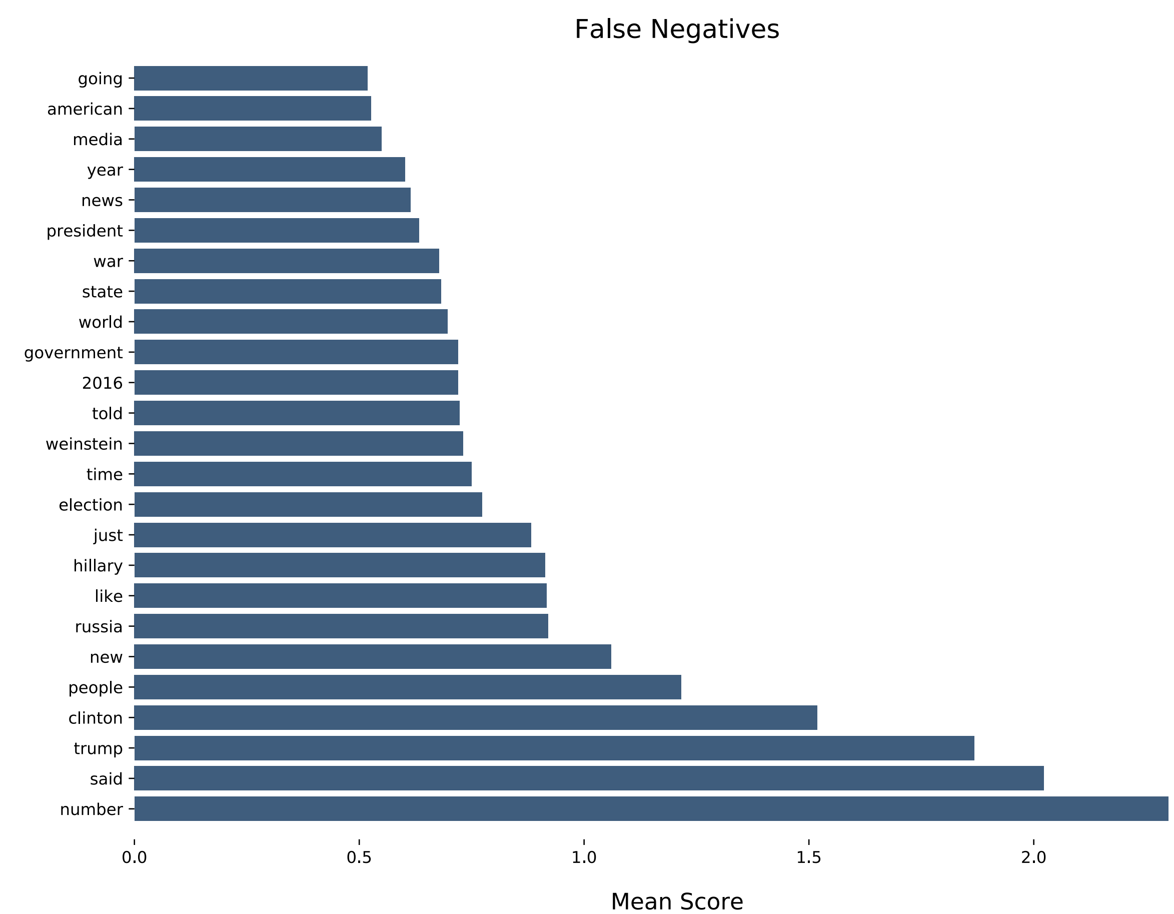
*** 13% Improvement in the specificity**

```
% Improvement: Optimized vs 1-gram
accuracy    : +3.864
f1score     : +2.367
precision   : +3.042
sensitivity : +1.692
specificity : +13.307
```



Understanding the Results

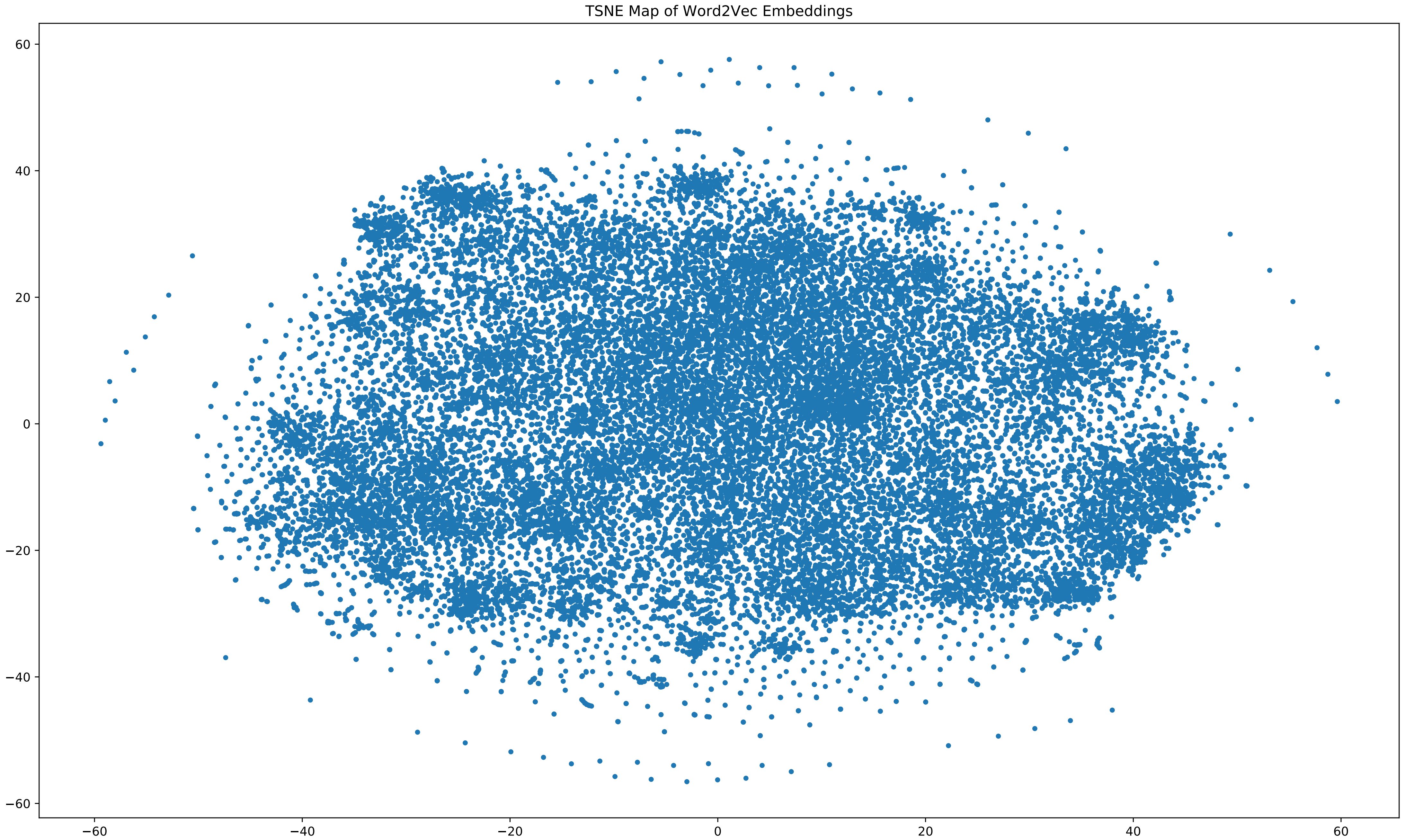




- Why did some articles fail to be classified correctly
- No discernible pattern amongst the most frequent words
- Many of the words in the failed classification are also the most common in the correct cases

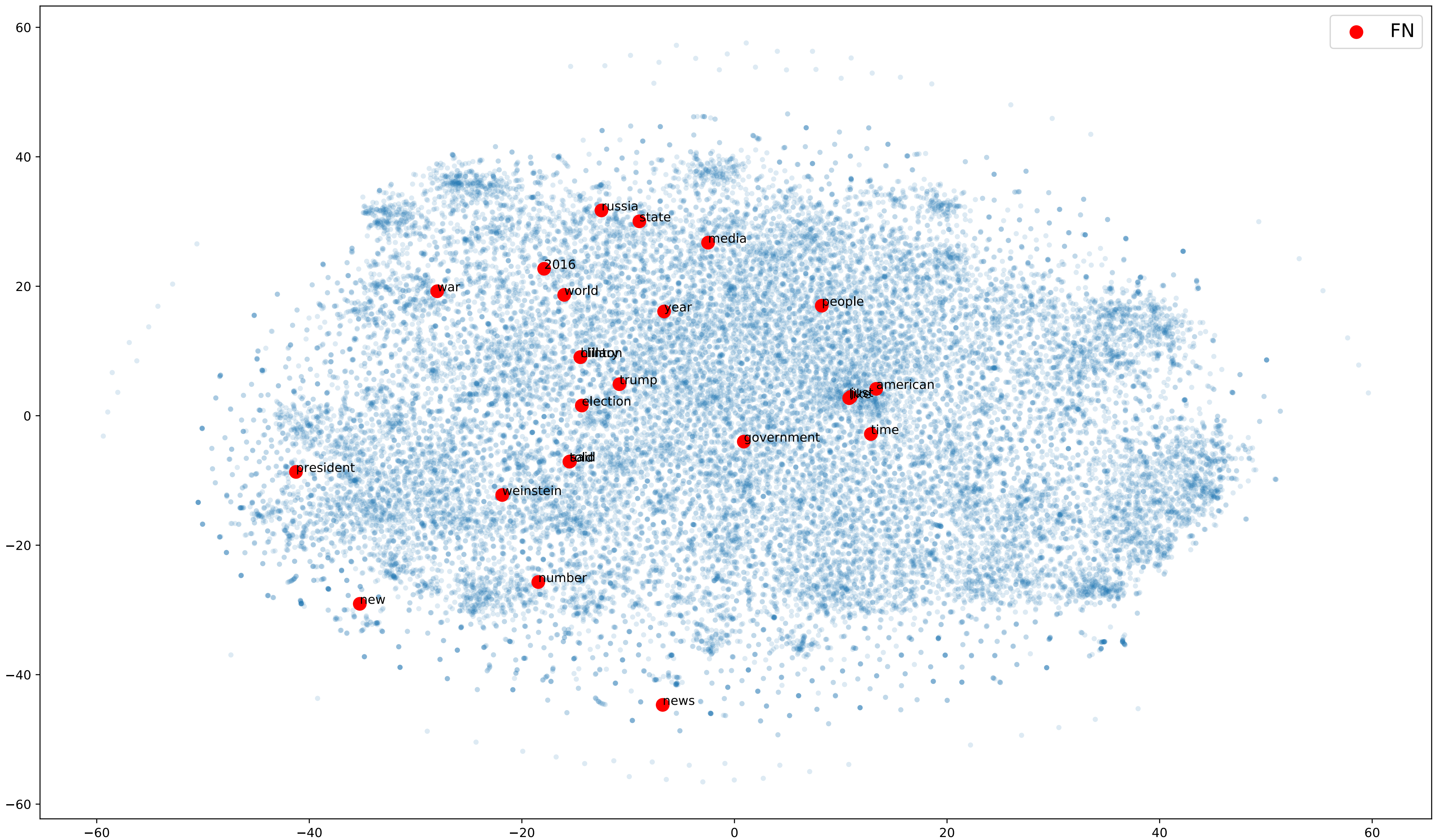
Looking at Word Relationships

- Word Embeddings with Word2Vec
  - Similar to n-grams in that we look at a words context
  - clusters of words are new features
    - based on words with high probability to be nearby in text



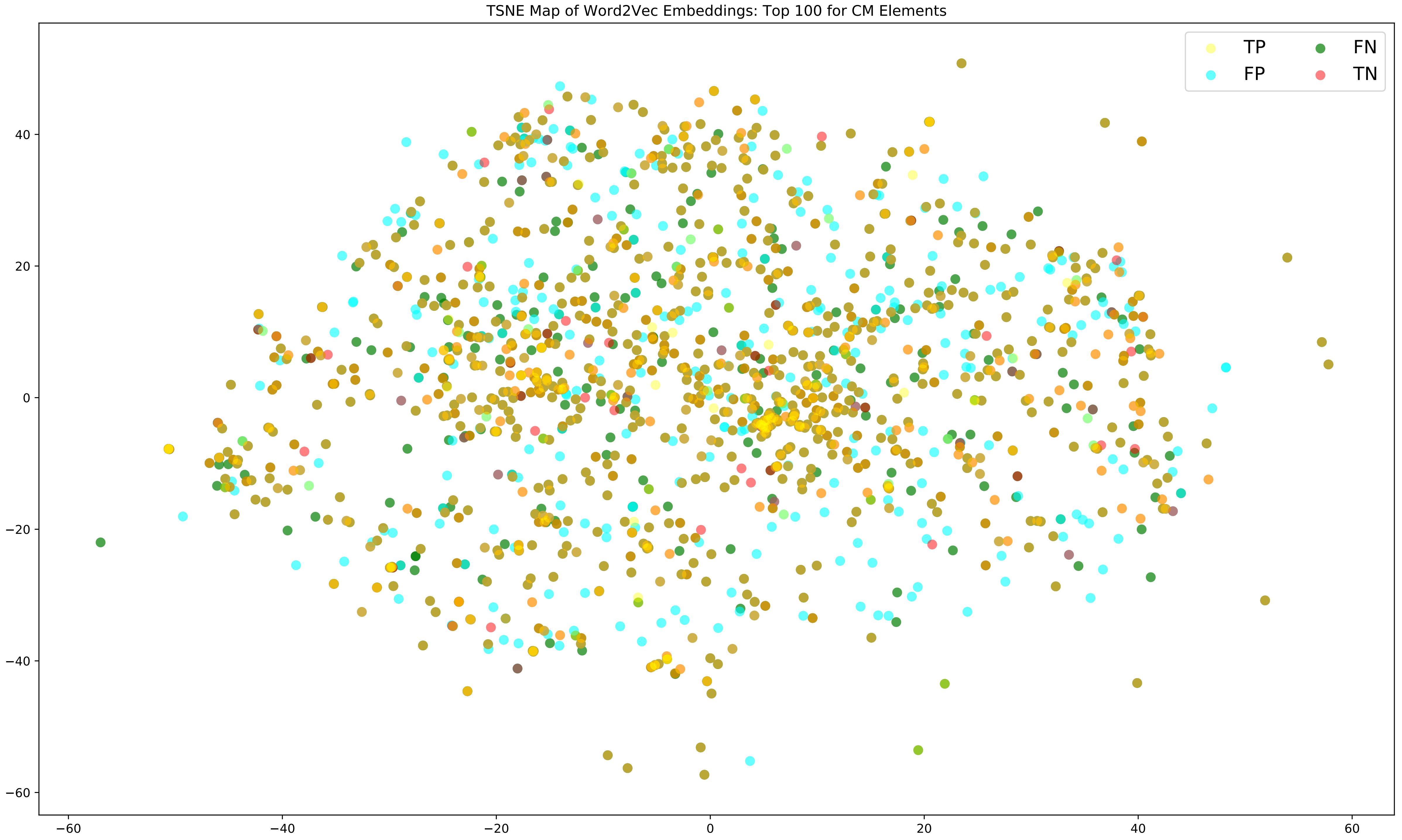
Why did we fail to flag some Fake News?

- Spread of most common words in FP articles exist mostly in separate clusters(features)
- This is expected since our classifier is operating at a high level of precision(98%)
- No obvious pathologies:
  - If we found FN words clustered tightly this would indicate an obvious correlation that the model failed to find



Top 1000 Words for All Cases

- Most frequent words are seeds for clusters that can be used as features in a new training



Conclusions/ Next Steps

- **An initial text classification analysis has been performed on Fake and Real news articles**
  - First Model: Bayes Naive Classifier(unoptimized, non-TFIDF, OOB)
  - 15,000 total article(~12,000 Fake and ~3,000 Real)
  - Performance: F1 Score = 0.96, Sensitivity = 0.96
- **Current Work/Next Steps**
  - Scraping additional Real news articles(see how performance changes as #Real -> #Fake)
  - Understand which words are most often associated with failed predictions
  - Hyperparameter Search of the Bayes Classifier(stop words, nGrams, Multinomial or Gaussian, etc)
  - Why do the TFIDF features create a biased/poor model?
  - Move to more complex modeling: RNN, word2Vec