Deception Detection:
Capstone 02 Milestone Report

Introduction

Pamela Meyer, author of Liespotting, takes the stage for her TED talk, “How to spot a liar”. I ready my pencil, eager to take down every morsel of knowledge she has to offer. Around 8 minutes into the talk, she uses the infamous Bill Clinton statement:

Figure 1: Bill Clinton's Infamous Lie (https://www.ted.com/talks/pamela_meyer_how_to_spot_a_liar?language=en#t-447811).

She then points to two indicators of deception: “non-contracted denial” (i.e., “did not” versus “didn’t”) and distancing language (i.e., “that woman”). Furthermore, she notes that “qualifying language” (i.e., “In all candor”, “To tell you the truth”, etc.), repetition of the question in its entirety, and too much detail are also telltale signs of deception.

I first saw this video in 2013 and it inspired my master’s thesis work: identifying the neural correlates of deception for gesture-based and speech-based lies. Flash forward to 2019 and deception still fascinates me. Now I’m wondering if I can find any other signs of deception using natural language processing. Better yet, can I make a machine learning (predictive analytics) model that can detect deception in speech? This led to three questions:

1. Can I extract a dataset of true and false statements to analyze verbal deception from the web?

2. Are certain patterns more common in either statement type (for example, less verbs, longer sentences, etc.)?

3. How well can machine learning differentiate truth from lie?

Purpose

Deception is plentiful, but research on the topic using machine learning is scarce. To fill this gap, data was scraped from https://www.politifact.com where primarily political statements are analyzed for veracity. The goal of this paper is three-fold: (1) create a refined dataset from this website containing true and false statements, (2) analyze whether certain patterns in speech are more common in either statement type and (3) use machine learning to predict deception. The target audience for this capstone is the media which could use the algorithm created from this dataset to create a "probability-of-deception" meter. Companies may also use this on candidate cover letters, etc.

Methods

Data Acquisition

Data was scraped from https://www.politifact.com using Beautiful Soup. Politifact is a not-for-profit national news organization which fact checks in a nonpartisan manner. Statements are found from transcripts, speeches, new stories, press releases, viral posts, and campaign brochures. Afterwards the statements are classified by three editors as follows: True, Mostly True, Half True, Mostly False, False, and Pants on Fire. Two out of three votes by the editors is all that is needed for final classification.

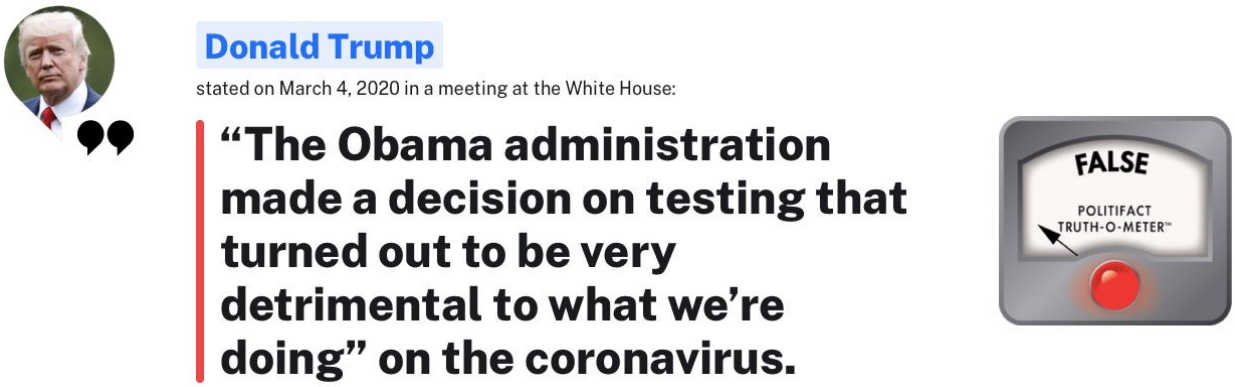

Figure 2: Politifact Layout featuring a lie told by Trump (https://www.politifact.com/factchecks/2020/mar/06/donald-trump/trump-wrongly-blames-obama-limits-coronavirus-test/).

Data Cleaning

The original scraped dataset contains 16,611 data points. For simplicity, this capstone will focus on statements that: (1) that are True or False, (2) which are direct quotes (direct quotes are defined as beginning and ending with a quotation mark), and (3) only sources with less than or equal to three statements were kept to maintain equal

representation of different speech patterns. After limiting these parameters, the dataset was left with 1147 data points.

Data Creation

During the exploratory data analysis over 100 columns were created with information such as part of speech tags, average word length, etc. leaving the final dataset with 1147 rows and 153 columns. The details for the creation of these columns follows.

Vocabulary qualities (count, length, readability score)

For each statement the number of characters, number of words, average word length, and a readability score were determined. For the readability score, the Gunning fog index was used. This "index estimates the years of formal education a person needs to understand the text on the first reading" (https://en.wikipedia.org/wiki/Gunning_fog_index). Broadly speaking, this index depends on sentence length and syllable amount to give scores ranging from 17 (College Graduate) to 6 (6th grade).

Part-of-Speech (POS) Tagging

To tag each part of speech for each word in each sentence automatically, spaCy was used. I chose spaCy because it is beginner-friendly and "[t]wo peer-reviewed papers in 2015 confirmed that spaCy offers the fastest syntactic parser in the world and that its accuracy is within 1% of the best available. The few systems that are more accurate are 20× slower or more" (https://spacy.io/usage/facts-figures). It makes predictions of which tag most likely applies to each word given its context. The model it uses is produced by showing an algorithm "enough examples for it to make predictions that generalize across the language – for example, a word following 'the' in English is most likely a noun" (https://spacy.io/usage/linguistic-features). Spacy has two systems (attributes) for POS tagging. The first tagging attribute, .pos_, is based on the Google Universal POS Tags (with some additions added by spaCy) and contains 16 tags. The second tagging attribute, .tag_, uses OntoNotes 5 from the Penn treebank tag set and contains 53 tags. Both systems will be used.

Syntactic Dependency

The library spaCy was also used to generate syntactic dependencies (i.e., relation between words). The syntactic dependency tags generated (48 in total) will also be examined. The figure below depicts syntactic dependency with arrows and POS tagging.

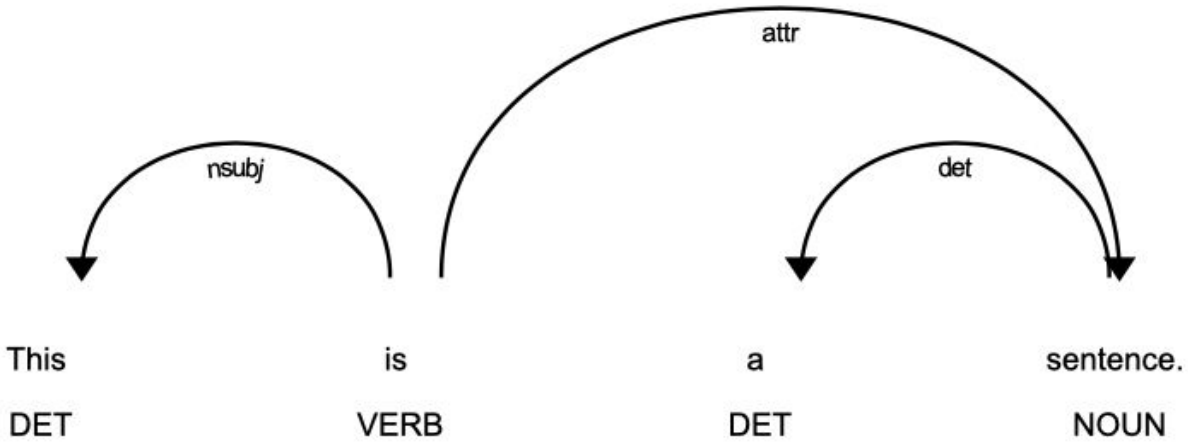

Figure 3: Part of Speech Tagging and Syntactic Dependency Visualized (https://spacy.io/usage/visualizers).

Named Entity Recognition (NER)

The library spaCy also "features an extremely fast statistical entity recognition system ... [which] identifies a variety of named and numeric entities, including companies, locations, organizations and products" (https://spacy.io/usage/linguistic-features#named-entities). It may also be the case that certain types of entities appear in deceptive statements or truthful statements with higher frequency. Therefore, NER tags (18 in total) will also be applied to the dataset. The figure below is an example of NER tagging.

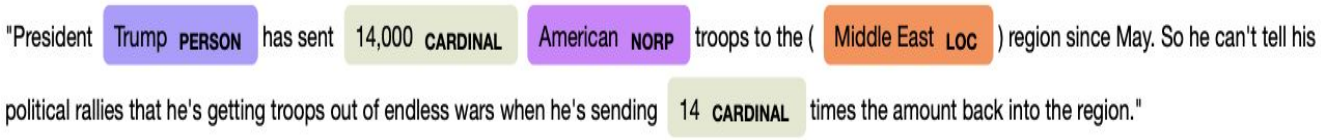

Figure 4: Syntactic Dependency Visualized (https://spacy.io/usage/linguistic-features).

Normalization

In natural language processing (hereafter, "NLP"), there are measures taken to ensure that only text with the greatest significance is analyzed. Typically these measures include the elimination or replacement of less meaningful items. This process is called text normalization.

One of these measures is lemmatization. Lemmatization takes a word and reduces it to the base form. For example, the base form of cars is car. "The goal of ... lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. For instance: am, are, and is become be ... [and] ...

car, cars, car's, and cars' become car" (https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html). By identifying the base form (i.e., the root form), it becomes easier to find the most frequently used words or concepts. Lemmatization is common practice in NLP.

Stripping html tags, removing accented characters, lowercasing all text, removing special characters (such as punctuation), replacing digits, and removing stopwords is also part of normalization. Stopwords are words that "occur most frequently if you aggregate a corpus ... like 'a', 'the', 'and'" (*Text Analytics with Python*, Dipanjan Sarkar, pg 154). That is, these words "have little or no significant value" (*Text Analytics with Python*, Dipanjan Sarkar, pg 154). Typically normalized text which contains fewer tokens (words, stems, etc.) is analyzed. In this case however, it is not known if stopwords, etc. are of value in deception detection, and therefore two branches of datasets are required for machine learning: one dataset will have light normalization techniques applied (stripping html tags, removing accented characters, and lowercasing.) and one will have all normalization techniques applied (all techniques mentioned above).

	Text
Original Statement	*"@ ! i DON'TTT, won't, a féél can't not USING the NLP 27x maaaah?"*
Lightly Normalized	*"@ ! i don'ttt, won't, a feel can't not using the nlp 27x maaaah?"*
Fully Normalized	*'feel use nlp maaaah'*

Table 1: Comparison of light vs full normalization.

Statistics

Statistics were created based on the raw text dataset, not the lightly normalized or fully normalized datasets since the spaCy taggers previously mentioned may not function correctly with normalized data. A simple t-test using scipy's statsmodel was conducted on each of the aforementioned created data comparing true versus false statements. Only the most interesting results are reported in this paper.

Machine Learning

The standard in natural language processing is to normalize text, i.e., distill content in statements by removing frequent words (i.e., stopwords), punctuation, etc. But, given the statistics from the data analysis revealed that the frequency of adverbs and determiners (i.e., stopwords), etc. varied with statistical difference in lies versus

truth, I hypothesize that normalization may lower the ability for machine learning algorithms to predict deception. To test this hypothesis, I will compare a lightly normalized permutation of the raw statements to a fully normalized permutation as shown in Table 1 above.

Vectorizers

Apart from the two datasets, there are several vectorizers for vectorizing text such as Bag of Words (hereafter, "BoW"), n-GRAM, and Term Frequency-Inverse Document Frequency (hereafter, "TF-IDF").

BoW

This is the process of turning text into a series of numbers (vectors) for machine learning models where each word is mapped to its frequency as follows:

	Text
Original	NLP is amazing; truly NLP is amazing.
Identify Unique Words	amazing, is, NLP, truly
Map to Frequency	vector 1 = [2, 2, 2, 1]

Table 2: BoW vectorization.

Of course, a vector will be much longer so that it can incorporate all words present in all texts. Whie BoW is a common vectorizer to learn first in NLP, it has one obvious disadvantage: words lose context because they are analyzed independently and without order.

n-grams

n-grams, acting as an evolution of BoW, map groups of words to their frequency thereby giving them dependence and thus context.

	Text
Original	NLP is amazing; truly NLP is amazing.
Identify Unique Word Groups	NLP is, is amazing, amazing truly, truly NLP
Map to Frequency	vector 1 = [2, 2, 1, 1]

Table 3: n-gram (size 2) model vectorization.

The above is an example of a size 2 n-gram. However, sizes can range such as 1-3, 4-4, etc. In general, sizes greater than 3 are not useful because n-grams of size 4 or greater are exceedingly rare (source: https://www.datacamp.com/courses/feature-engineering-for-nlp-in-python). Therefore, 2 n-gram sizes will be used: 1-2 and 1-3. Note that an n-gram size of 1-1 is a BoW model.

TF-IDF

This vectorizer further expands on the n-gram vectorizer, but additionally scales terms based on their frequency throughout all your text. The intuition behind the scaling for TF-IDF is as follows: if a word appears often in a document it may be important (TF - term frequency), but if a word appears often throughout many documents, then it just might be a common word that does not add much value (IDF - inverse document frequency).

Collectively, there will be 4 vectorizers examined in this capstone: BoW, n-gram size 1-2, n-gram size 1-3, and TF-IDF of n-gram size 1-2.

Classifiers

Three classifiers will be used for binary classification: Multinomial Naive Bayes, Logistic Regression, and Random Forests. I've chosen these algorithms for various primary reasons: Multinomial Naive Bayes is a typical baseline algorithm for NLP while Logistic Regression and Random Forests are the top two most used algorithms according to "Kaggle's State of Data Science and Machine Learning 2019" survey of data scientists.

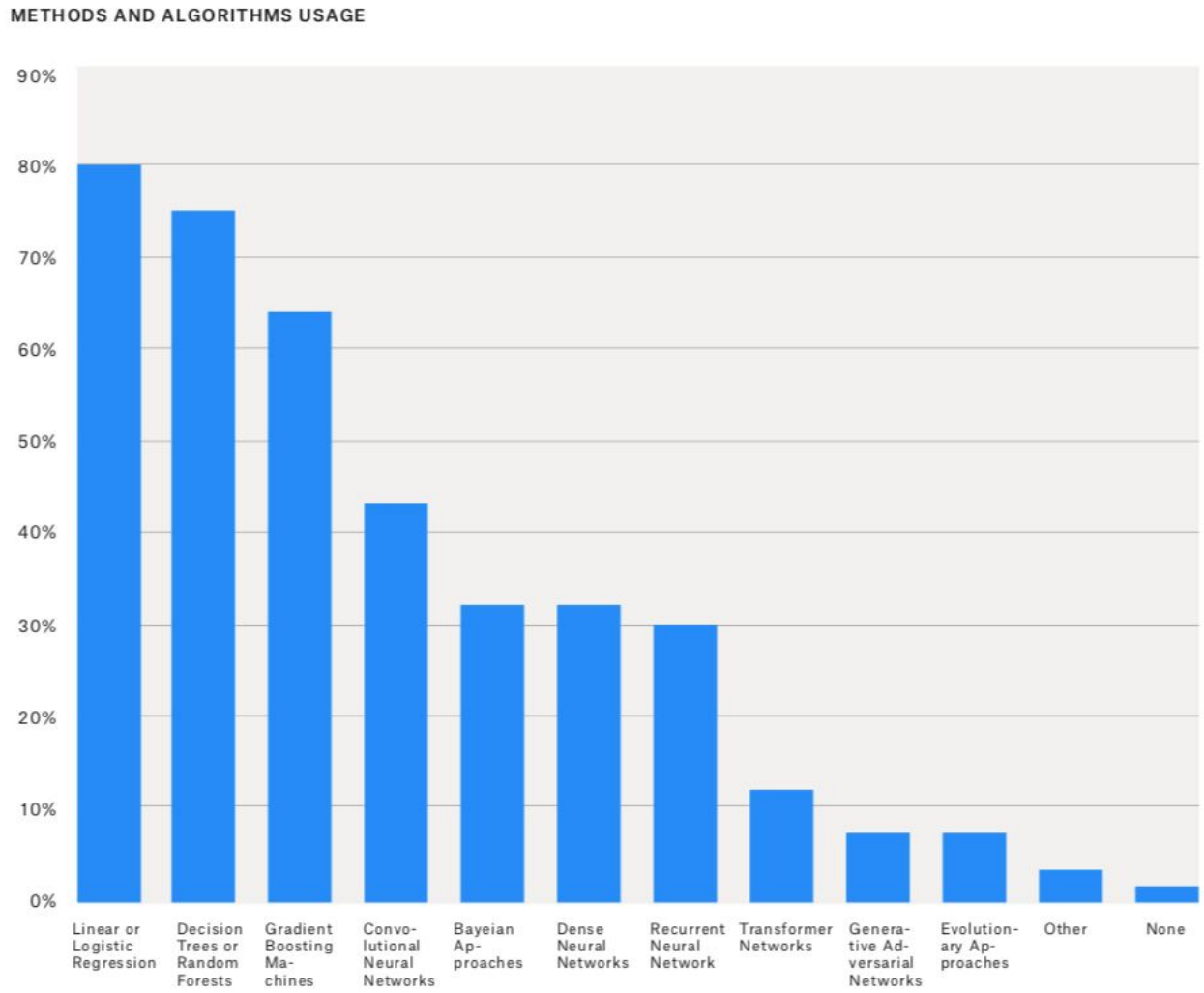

Figure 5: Most used algorithms according to "Kaggle's State of Data Science and Machine Learning 2019" survey of data scientists.

In addition, Random Forests have non-linear decision boundaries whereas the other two algorithms have linear decision boundaries which can help determine which boundary best fits the data.

Validation and Optimization

I split the data into a train and hold-out set. The train set was grid-searched and cross- validated (n_folds = 3) and optimized using accuracy because the data was balanced (50% lies and 50% truth).

Results and Discussion

Statistics

Vocabulary Qualities

Character count and word count are statistically higher for truthful statements (p = 0.042), and average word length is statistically higher for falsehoods (p = 0.003). That character/word count is lower for liars is in line with research that found those under stress speak less than when not stressed (https://www.nature.com/articles/nature.2017.22964). As for average word length being higher for liars, perhaps liars rely on a vocabulary of larger words to convince others of their lie, i.e., they may be overcompensating. One caveat to this hypothesis is that the Gunning fog readability index score (primarily measuring sentence length and syllable count) was not statistically significant. Perhaps because this index acts similar to a text normalizer by not including common suffix endings such as -es, -ed, or -ing in its algorithm, it inadvertently destroys statistical significance.

Part Of Speech Tagging (hereafter, "POS")

There were two POS tagging systems used on this dataset. For simplicity, only the simplest results from the first tagging system, .pos_, are discussed here:

	Mean Number in ***Truthful*** *Statements*	*Mean Number in* ***Deceptive*** *Statements*	*p-value*
Adjective	***1.42***	*1.23*	*.006*
Noun	***4.35***	*3.95*	*.002*
Verb	*1.72*	**1.89**	*.040*
Number	***0.99***	*0.72*	*.000*
Adverb	***2.33***	*2.05*	*.002*
Determiner	***1.87***	*1.64*	*.006*

Table 4: Mean number of part of speech tags in truth and deception comparing their respective p-values.

Adjectives, adverbs (i.e., "of", "about", "to", etc.), determiners (i.e., "a", "an", "the"), nouns, and numbers appeared statistically more often in truthful statements.

Interestingly, it has been reported that adjectives and adverbs occur more frequently when people are under stress (https://www.nature.com/articles/nature.2017.22964). This suggests that the way stress affects speech and the way lying affects speech may differ. With regard to adverbs and determiners, it is of value to note that those two parts of speech are typically removed during text normalization as they are stopwords. However, keeping them may help when detecting deception. Finally, numbers may be more common in truth since reporting dates, percentages, etc. may be something a truthful person is more likely to do.

False statements, it was found that verbs appear statistically more frequently. Perhaps liars are more focused on words that reflect a chain of events rather than describing the event with detail-oriented words such as adjectives and adverbs. That is, truth-tellers appear to use more descriptive and sensory language while liars use only the language necessary to communicate a falsehood they hope an audience will believe.

Named Entity Recognition (hereafter, "NER") Tagging

NER tags gave the following results:

	*Mean Number in **Truthful** Statements*	*Mean Number in **Deceptive** Statements*	*p-value*
People	*0.19*	***0.29***	*.003*
Region	***0.46***	*0.38*	*.043*
Date	***0.39***	0.27	*.001*
Percentage	***0.16***	*0.10*	*.007*

Table 5: Mean number of NER tags in truth and deception comparing their respective p-values.

References to people are statistically higher for liars than truth-tellers. Since Politifact is a website that primarily examines political statements, and politicians engage in false smear campaigns, it follows that references to others are common in false statements. References to dates and percentages appearing more frequently in truthful statements is logical since it is more difficult to lie about a specific reference to time or to data. Region references occuring more often for truthful statements is a unique insight only NER could have found. Perhaps it is less profitable and more difficult to tell a lie about a region than about a person.

Machine Learning

There were 2 datasets, 4 vectorizers, and 3 classifiers which leads to a total of 24 models. Only the most unique results among these models will be discussed.

The cross-validated accuracy score on the best training set was 63% - the light normalization/TF-IDF/Logistic Regression model. That model's accuracy on the holdout set was 57%. Interestingly, the model was much better at predicting truthful statements than it was at predicting deceptive statements with recall of .65 and .50, respectively.

Out of curiosity, I checked how the other models performed on the holdout set and no model performed better than 58% accuracy.

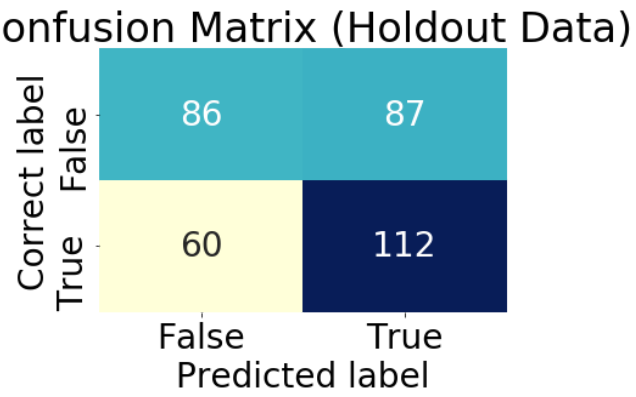

Figure 6: Confusion Matrix results for the light normalization/TF-IDF/Logistic Regression model on the holdout set.

When comparing the light normalization models versus the fully normalized models, the fully normalized models only performed better for the BoW vectorizer. For all other vectorizers, fully normalized datasets underperformed for the cross-validated training and holdout set accuracy. Likewise, Random Forests always underperformed against their linear-decision-boundary counterparts.

Conclusion

Statistics

For truthful statements, statistically higher frequencies of the following occur: character count, word count, adjectives, adverbs, nouns, numbers, and determiners. For false statements, statistically higher frequencies of the following occur: average word length, verbs, and references to people. It may be the case that liars rely on a more complex vocabulary centered on action-based words because they lack a true experience. Conversely, their truthful counterparts appear to add descriptions and thus are more verbose in their prose.

Machine Learning

The statistical analysis revealed that stopwords were used with significantly more frequency by truth tellers. It was then found that lightly normalized text delivered higher accuracy scores implying that the removal of stopwords is not a fruitful endeavor in deception detection.

Unfortunately, accuracy scores on the holdout set were all below 60% (maximum of 58%) implying that the models chosen may not have been the best choices or that deception is much more complex than any model can capture.