**Alternus Vera: A Fake News Detection Model**

Golnaz Bidabadi (Student ID: 008022755)

***Note: In this document, I have documented the three factors that I individually worked on to analyze and determine the impact of each factor on fake news detection.***

### ***Abstract - Fake news is one of the most critical societal concerns today, as it potentially plays a pervasive role in manipulating opinions and deceiving individuals and thus introduces a profound impact on society. The rise of social media as one of the main resources for news consumption as well as everyday use of digital devices by increasing number of people have contributed significantly to propagation of the fake news. Therefore finding ways of detecting and assessing the truthfulness of news is one of the most-needed solutions to minimize the impact of deceptive news claim. This paper proposes a solution using natural language processing (NLP) techniques and Machine learning algorithms to help achieve maximum accuracy in fake news detection.***

**Keywords**— fake news detection, veracity, Natural Language Processing, Social Media, Machine learning, Ensemble methods, sentiment, Latent Dirichlet allocation (LDA)

# **Introduction**

Today we live in a world where there are new terms that have been coined like “alternate facts” where in fact a fact is defined as a statement that holds true. Fake news has engulfed the world that we live in. Fake news is not only used to spread propaganda but it is also a term that is used to discredit sources that convey facts that a speaker doesn’t necessarily agree with.

## FAKE NEWS DEFINITION

## The first step in trying to detect fake news is to define what fake news actually is. Fake news is something that doesn’t agree with the facts. Facts can represent a certain historical occurrence, a scientific truth or a documented statement like the constitution or the law of the land. One of the biggest challenges in today’s world with so much information being circulated is how does one identify real news versus fake news. This paper tries to document an effective mechanism by which fake news can be discerned.

* 1. FAKE NEWS DETECTION

Fake news detection is defined as the prediction of the chances of a particular news article (news report, editorial, expose, etc.) being intentionally deceptive by Rubin, et al. (2015). The paper attempts to detect fake news in a methodological approach toward categorizing news in a spectrum of veracity, trying to measure the fakeness of news articles. It takes into account the body of the article, the headline (title) of the news as well as its sources (author[s], speaker[s] and publisher[s])

# Methodology

The aim of the project is to assess the veracity of a particular news article. Considering Shu, et al. (2017) paper, one of the first steps that we need to take for detection of fake news is to identify the main factors that would be considered in the computation of the fake-news-likelihood scores. After identifying the most important factors, various NLP and machine learning techniques were implemented, and extensive analysis was performed to determine the impact of each factor in detection of fake news. Detailed implementation and analysis of all the considered factors are demonstrated in the section 2.2.

* 1. Data Enrichment and Datasets

To tackle the fake news problem we used an ensemble method. Ensemble is a technique that creates multiple models and then combines them to produce better results than any of the single models individually.

In this project for computing different aspects and factors of the model we used alternative sources of data. Each factor was trained on different datasets based on its characteristics. The collective model which is the polynomial equation that we develop at the final stage of the project has been trained on these distinct datasets, each of which relate to a particular factor. We have then defined specific outputs for the equation. Therefor, when a new test dataset comes in as the input into the model, the polynomial equation will classify the incoming dataset as one of the output labels by looking at the aspect of the input dataset. The different datasets used for factors are introduced below.

## Liar Liar, pants on fire

The dataset is collected through the API of the PolitiFact website is a fact-checking US-based website. The work has been done by Wang, W. Y. (2017) in their benchmark dataset called “Liar Liar Pants on Fire” to come up with a data set that can help to learn how real news can be distinguished from fake news. The author defines a range of veracity (*'true'*, *'mostly-true', 'half-true', 'barely-true', 'false', and 'pants-fire'*) to determine the fakeness of a statement. The study identifies fakeness by looking at the historical track record of the speakers in their statements. How many times did the person speak the truth? How many times did the person’s “pants were on fire”? How many times was the person half true? These can determine if the person would speak the truth the next time or not. The authors allude to the fact that there are many other mechanisms to detect fakeness. Some of which are discussed in this paper.

## Fake News Challenge

The goal of the Fake News Challenge is to overcome the problem of fake news by using Machine Learning and NLP techniques to assess the veracity of a news. It focuses on the problem of *Stance Detection* to try to automatically assess the stance of the body of a news article with respect to its headline. The assumption behind this approach is that the news is likely to be genuine if multiple credible news sources show a positive stance for similar headlines. On the other hand, the veracity of the news is questionable if sources with less credibility show a positive stance toward that news. The data provided in the fake news challenge dataset are headline, body, and stance instances, where stance is classified as one of *agree* (The body text agrees with the headline), *disagree* (The body text disagrees with the headline), *discuss* (The body text discuss the same topic as the headline, but does not take a position), *unrelated* (The body text discusses a different topic than the headline)

## Kaggle Dataset: Getting Real about Fake News

Getting Real about Fake News is a Kaggle dataset contains text and metadata from fake and biased news sources around the web. URLs are collected from BS Detector which is a plug-in used by Mozilla and Chrome browsers to detect unreliable sites and sources and to warn the user accordingly. BS Detector just states a warning message if the article is found to be fake It does not specify the percentage of error and neither does it classify news into levels of “fakeness”. The types in the Kaggle dataset are labeled as one of *bias*, *fake*, *conspiracy*, *bs*, *satire*, *hate*, *junksci*, *state*.

2.2.1. **Lexical Features Credibility**

One of the well-known principles of design is KISS ("keep it simple, stupid") which emphasizes on simplicity rather than making the system complicated. Motivated by the KISS principle and following the paper published by Shu et al. (2017), which recommends the linguistic factors as one of the features that can be used for fake news detection, we studied to determine if the lexical features could be useful for detecting the truthfulness or fakeness of news.

1. **Data Preparation and Exploratory Data Analysis (EDA)**

The Liar Liar dataset is divided into three sub-datasets: training dataset, test dataset and validation dataset. All datasets are in tsv format. The step taken for data preparation and data analysis are as follows:

1. Adding the column names to each of the sub-dataset: The name of columns was extracted from the readme files and were added to all sub-datasets
2. Data analysis based on the factor, credibility of lexical features: After a basic data analysis, two columns were identified that needed to be processed during the experiment:

* The “label” column, which has been label by human annotators and includes the level of the veracity of the data records.
* The “statement” column, which consists of the short publish news (claim) written in natural language (i.e. English).

1. Exploration of the quality of the dataset: To make sure there are no missing values in the dataset the target columns were explored.
2. Data Analyzation of values in the target columns: Since the statement column includes no missing values and also include natural language sentences, no further value analysis on this column seemed to be necessary. However, A data value distribution analysis on the label column was done as demonstrated in the below table.

|  |  |  |
| --- | --- | --- |
| **Distribution of label-classes in the training dataset** | **Distribution of label-classes in the test dataset** | **Distribution of label-classes in the validation dataset** |
|  |  |  |

**Table - Distribution of label-classes in the three sub-datasets**

As shown in the table, all datasets contain of different types of label-classes. Therefore, all classes should be considered in the classification process and none of them should be dropped.

1. **Data Enrichment**

Considering the best practice of fake news detection, discussed by Shu, et. al (2017), the following lexical features were chosen to be extracted from the statement values:

* Total number of words,
* Average number of characters per word (average word-length)
* Frequency of large words
* Frequency of unique words

As part of the data enrichment, the total number of words was added as a new column to the sub-datasets. Next the average number of characters per word in each statement was added to the sub-datasets. Then then the frequency of large words in each statement was added. Based on the average length of the words that was observed, large words were defined as a word which has 9 or more than 9 characters. Finally, the frequency of unique words in each statement was added to the sub-datasets. A word has been defined as unique, if it has been used in one and only one of the statements.

1. **Feature engineering and Data Preprocessing**

The data we have is all in text format. Therefore, preprocessing was done because some sort of numerical feature vector is needed to perform the classification task.

* Removing numbers and punctuations
* Stemming
* Creating the Bag-of-words model

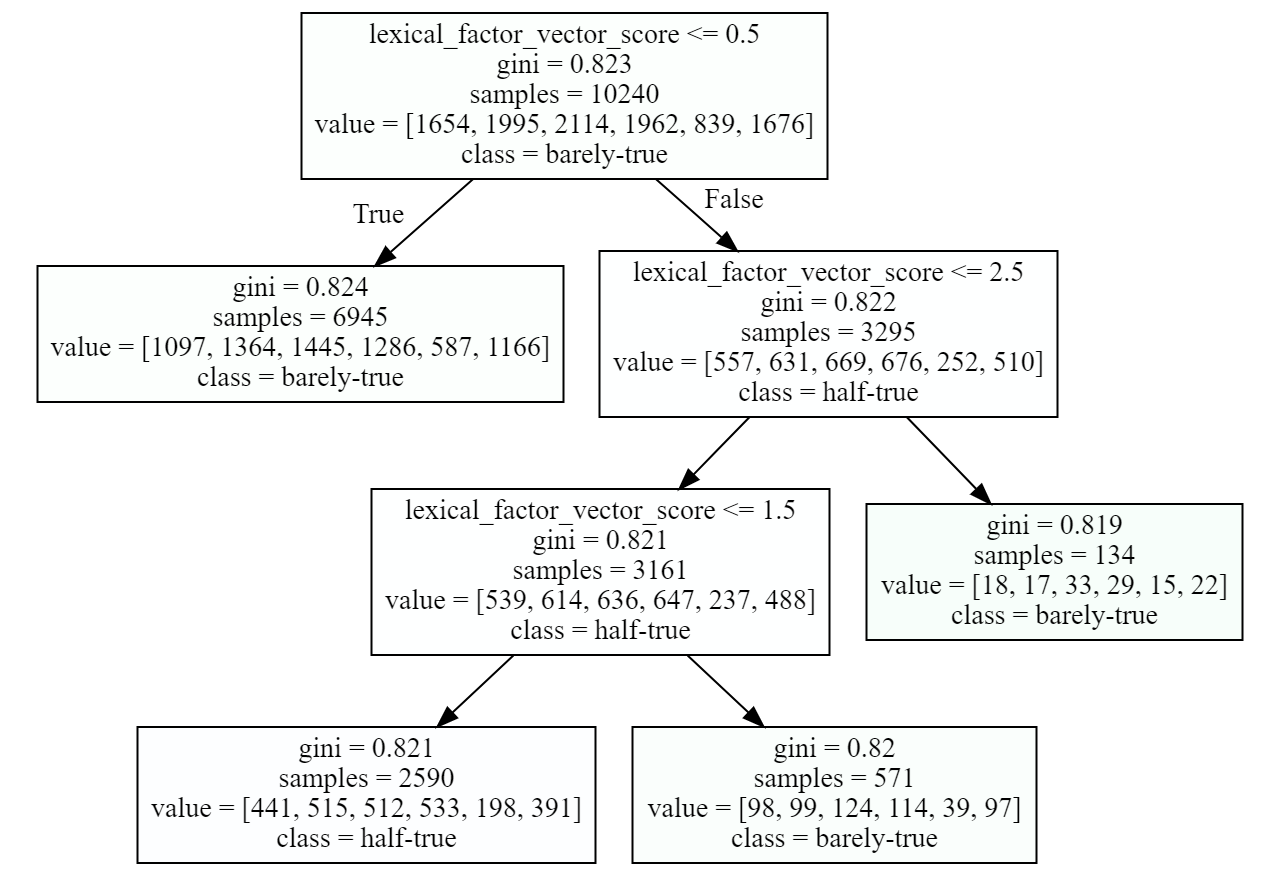
For the first three lexical features (out of four) stemming should have been done as the goal was to extract those features from the text as it was. Feature four (Frequency of unique words per statement) needed preprocessing, including removing numbers and punctuations, conversion to string, stemming as well as the bag-of-word model. A word has been defined as unique, if it has been used in only one of the statements.

Classifications and Evaluation of Classifiers

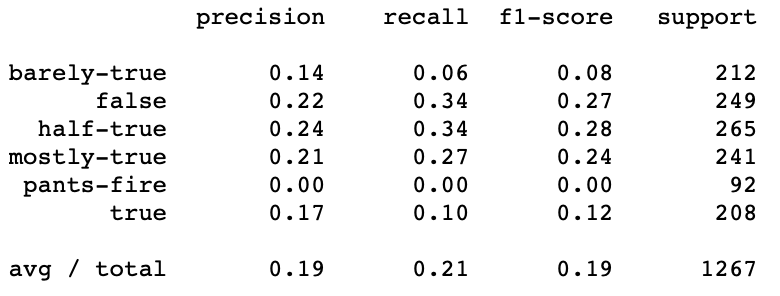
The lexical feature credibility factor is a vector of four extracted lexical features as a its dimensions. Below approaches were applied on the factor.

Decision Tree Classifier based on the lexical features

Decision tree classification was implemented on the lexical features as a vector of four features.



The accuracy score was 21.468034727703238. The evaluation results of the decision tree fitted classifier is shown below.



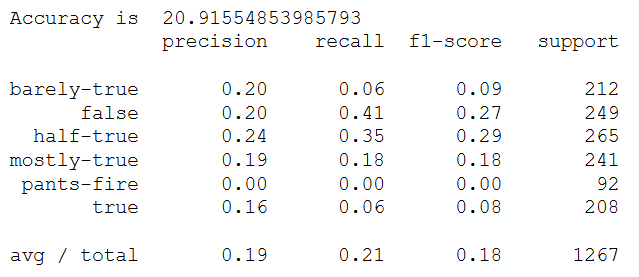
Different max\_depth and min\_samples\_leaf were tested and evaluated in the process of setting up the classifier. The above setting which was reported in this document was one of the best possible variants. Moreover, the classifier was also evaluated by subsets (the four features) of the extracted lexical features. However, none of the subsets resulted in a better accuracy than the above reported one.

1. **Mapping the lexical features vector to a scalar feature and running the Decision Tree classifier.**

Four lexical features were extracted, each of them was added as a separate column to the dataset and the classifier was trained based on that feature. The lexical features was considered as a vector consisting of 4 features. The assumption, however, was to let the classifier infer the importance of each feature. In the final polynomial equation for calculation of fake-news-likelihood a single score of the factor lexical-feature is needed. Therefor, a function was calculated and developed to map the the lexical features vector to a scalar factor. The developed function is as follows:  
*Def lexical\_factor\_vector\_score( word\_number, average\_word\_length, large\_words\_frequency, unique\_words\_frequency):*

*return(word\_number/10+average\_word\_length/5+large\_words\_frequency+9\*unique\_words\_frequency)*

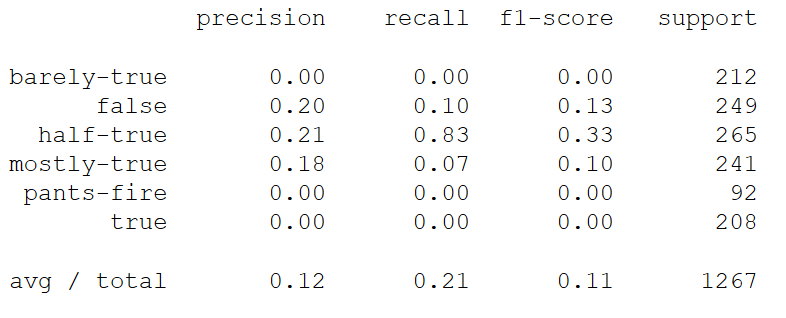
The above function shows the polynomial equation that was used for calculation of the lexical features factor. word\_number and average\_word\_lenge are normalized. All features except unique\_words\_frequency has given a weight of 1, while the unique\_words\_frequency feature has a weight of 9. The weightings were identified by more than 100 times different try-and-error training of a decision tree classifier. In each try different weights were associated to each factor to find out the optimal weightings. One important observed result here was that out of the four features created for lexical features “frequency of unique words” has the most impact on the detection of fake news. As a result, the highest weight was assigned to this feature in the equation. The results of the classifier evaluation is presented below.



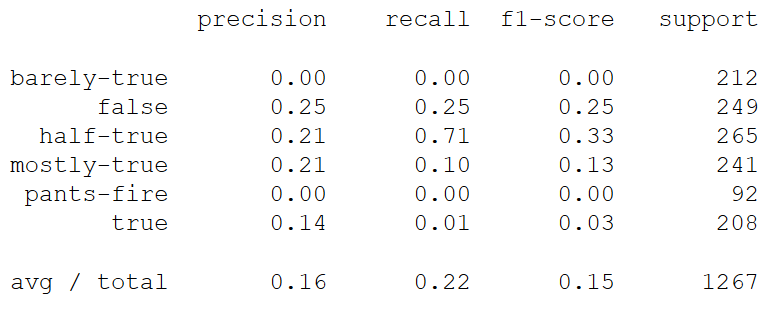
1. **Classification using Naïve Bayes Classifier based on the lexical features**

The results of the two previously discussed approaches were not satisfactory. As such the naïve Bayes classifier was also tested to see if it would improve the result. Again both lexical features as a vector and lexical features as a single scalar factor were used in training the classifier.

The evaluation results of the Naïve Bayes classifier on the scaler-feature-based factor is as follows:



The evaluation results of Naïve Bayes classifier on the vector-feature-based factor is demonstrated below.



As you can see the general results of the naïve Bayes classifier on the lexical features are not satisfactory and are similar to the result of the decision tree classifier. One interesting observation was that the recall score of the *half-true*category is very high in this classifier. However, considering the low precision, it would not be very helpful in general.

1. **Results and Analysis**

The evaluation results shows that precision and recall score of the classifiers (decision tree and Naive Bayes) for the factor of “lexical features credibility is not very high. Therefor, in the final polynomial equation of fake news detection a low weight should be assigned to this factor. But this study showed that basic lexical features could be as useful as some other high-level factors in detection of fake news when it is considered in aggregation with other factors. This can be seen as an important finding of the research done on this factor, which once again shows that KISS principle should always be considered in design.

**2.2.2. Speaker reputation**

While the history of humankind is the history of lies and truths, the concept of “fake news”, in the way that is used nowadays, is a very new concept. The fake news entry in Wikipedia has been only created in January 2017[[1]](#footnote-1). There is no doubt that the champion of the socio-political discourses surrounding the concept of “fake news” is Donald Trump. Considering this example, it is fair to assume that the speaker of a news (or the person who is referred in the news) can influence the chance of fakeness. Based on this assumption, we studied whether the prior reputation of the speaker could be used as a factor for predicting the fakeness of a news. The assumption here is that if many fake (or true) news has been published about a speaker, the probability of having the same type of news about that speaker in the unseen news of dataset is higher.

1. **Data Analysis and preparation**

The Liar Liar dataset was used for this factor. To work on the speaker reputation as a factor in detection of fake news the available information provided by the Liar Liar dataset was explored. The following columns could be used for investigating the speaker reputation in the dataset:

*'speaker'*, *'speaker\_job'*, *party*, *barely\_true\_counts*, *false\_counts*, *half\_true\_counts*, *mostly\_true\_counts*.

Features that were left out in analysis of speaker credibility in this study are as follows:

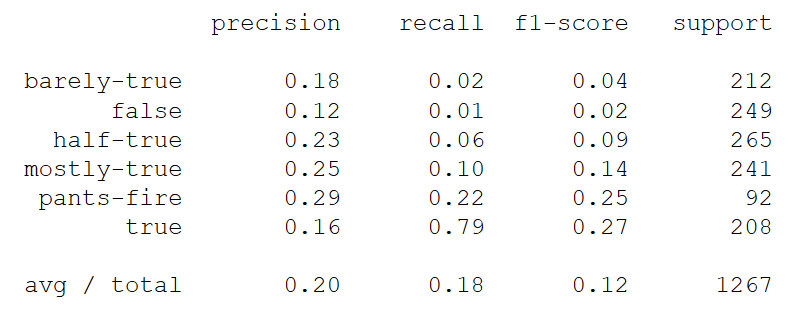
* ‘speaker’: In general, the speaker’s name could be an interesting feature for determination of speaker credibility. However, in Liar Liar dataset, most of the speakers appear only once or a very few times in the dataset. As a result, it does not seem that the speaker’s name would be a very suitable candidate, considering the dataset.
  + The same argument regarding the speaker also hold for the speaker\_job.
* ‘party’, or political affiliation, is a very good candidate to be explored However, since the political affiliation factor was analyzed as a separate, ‘party’ as one of the features was not included here.

As a results, only the history of the speaker as provided by the dataset was considered to infer the veracity of news and fake news detection. The barely\_true\_counts, false\_counts, half\_true\_counts, mostly\_true\_counts columns were used to explore the factor. The hypothesis here is that, it might be possible to infer fake news based on the number of fake news that have been assigned to the speakers previously. Clearly, this does not mean that a speaker, is a person who produces the fake news or that every single news about a particular speaker is fake. However, what could be inferred from the assumption is that the possibility that a fake news is produced about a particular speaker could be higher than others. For example, it might be the case that the number of fake news that are generated about Obama are more than Trump. Therefore, the history of (the number of) fake- or non-fake news about a speaker is considered in determining the impact of the factor in fake news detection.

The same data preparation steps as explained in section 2.1. were applied.

1. **Classification and Evaluation of the classifier**

Naïve Bayes classifier was used to classify fake news based on the provided columns that was explained previously. The result of the classifier to detect fake news based on a vector-feature definition of the speaker credibility are shown below. The vector includes the *barely\_true\_counts*, *false\_counts*, *half\_true\_counts*, *mostly\_true\_counts* columns.



As we can see the results are not satisfactory, which is not surprising. The main reason as discussed earlier is that not all statements of a speaker, e.g. Barack Obama, are fake or non-true. But still this factor could be considered in the final polynomial equation, as the probability of a news to be fake could be influence by the speaker of a statement.

# Deep Learning/Neural Networks classifier was also applied on the factor. It was done as an experiment to test whether it could improve the result in classification of news based on the speaker reputation features. Based on the tested settings, the achieved result was not satisfactory as shown below.

# 

1. **Analysis and comparison between lexical feature credibility and speaker reputation**

None of the factors that were tested so far could be individually used to detect all classes. As a result some comparison between lexical features credibility and speaker reputation was done. Other approaches were also tested in a try to improve the results.

1. **Comparing predictability of single label-classes based on the lexical features and speaker-reputation (history) features using one-hot-encoding**

A comparison between every single class of labels was conducted considering it might provide novel insights. Moreover, it was assumed that it could be the case that if the classifier was trained only for predicting one of the label categories, this could improve the results. For this purpose, one-hot-coding was applied on the labels, and naïve Bayes classifier was trained to predict every single class that we have in the dataset. In the following, the results of this process for each of the classes based on lexical features and speaker-history features is compared.

|  |  |
| --- | --- |
| **Evaluation results based on the lexical features => Label: true** | **Evaluation results based on the speaker-history features => Label: true** |
|  |  |

**Reflection:** Lexical features is not suitable to be used for detecting the ‘true’ news. But speaker-history produces a 29 percent precision for detecting the ‘true’ news.

|  |  |
| --- | --- |
| **Evaluation results based on the lexical features => Label: mostly-true** | **Evaluation results based on the speaker-history features => Label: mostly-true** |
|  |  |

**Reflection:** Both Lexical features and speaker-history features produce a very bad recall for the mostly-true news. The precision of the lexical features is better.

|  |  |
| --- | --- |
| **Evaluation results based on the lexical features => Label: half-true** | **Evaluation results based on the speaker-history features => Label: mostly-true** |
|  |  |

**Reflection:** Both Lexical features and speaker-history features produce very bad recall for the half-true news. The precision of the lexical features is better.

|  |  |
| --- | --- |
| **Evaluation results based on the lexical features => Label: barely-true** | **Evaluation results based on the speaker-history features => Label: barely-true** |
|  |  |

**Reflection:** Lexical features is very disappointing on the barely-true labels, and speaker-history features seems to be a better feature for this class.

|  |  |
| --- | --- |
| **Evaluation results based on the lexical features => Label: false** | **Evaluation results based on the speaker-history features => Label: false** |
|  |  |

**Reflection:** Both Lexical features and speaker-history features produce a very bad recall for the false news. The precision is almost the same.

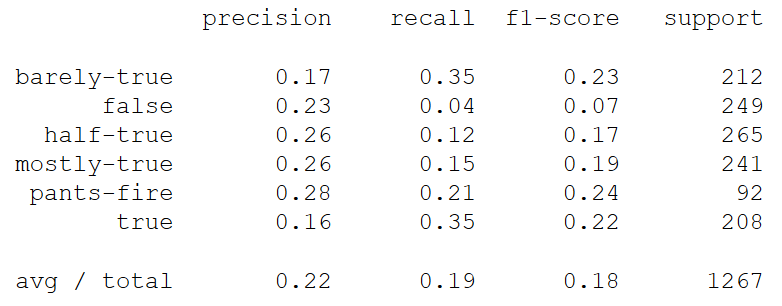
|  |  |
| --- | --- |
| **Evaluation results based on the lexical features => Label: pants-fire** | **Evaluation results based on the speaker-history features => Label: pants-fire** |
|  |  |

**Reflection:** The lexical-features cannot predict the pants-fire news. Although, the speaker-history features also has a very bad recall.

It seems that if we train the classifiers only based on single target labels for the lexical features credibility factor and speaker reputation factor it can only be useful for detecting ‘true’ news. This is not a surprising result, since enough information was not provided for the classifier in this scenario. However, the importance of this finding is that it clearly shows that if we consider the truthfulness, falseness, or fakeness of news as a relative concept which cannot be dealt with as a binary classification problem, then solving the problem in a binary way would not be helpful to understand the gradual changes in the classes.

1. **Combining the two factors of lexical features and speaker reputation**

Both lexical features and speaker reputation factors will be apply separately in the final polynomial equation that will be used to detect fake news. But as an experiment, a naïve Bayes classifier was trained and evaluated on the combination of the vector-features of both factors. The result is as follows:



The result shows that aggregating two factors is not far better than the results of the each factors separately.

**2.2.3. Title-Body-Correlation**

# The stance correlation between body and headline of a news has been shown to be an appropriate factor to detect the fakeness level of a news (e.g. see Thota 2018). However, there have not been much study to test whether the correlation of other features of titles and bodies, could be helpful factors for predicting the type (level of fakeness/truthfulness) of news. Therefore, it was tested, whether Doc2Vec, LDA, and sentiment correlation between titles and bodies of news could be useful factors for detection of fake news.

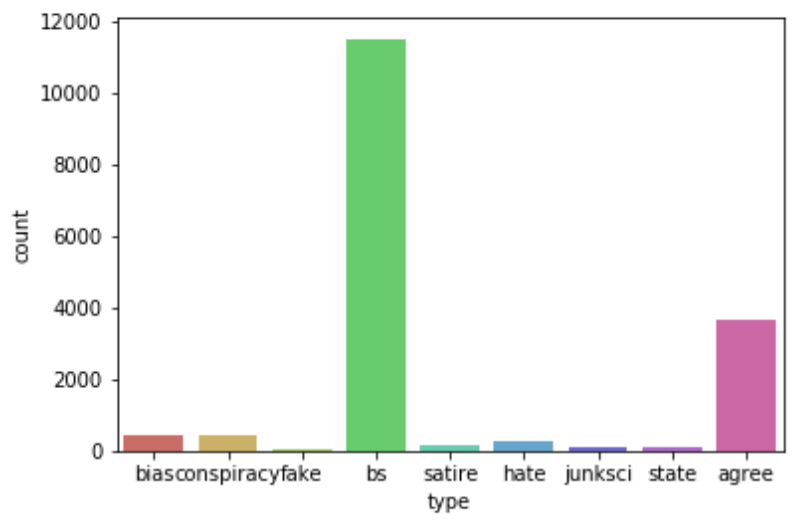
# **Data Enrichment**

The two “Getting Real about Fake News"[[2]](#footnote-2) Dataset and "FakeNewsChallenge"[[3]](#footnote-3) dataset were combined to enrich the dataset to see the impact of the title-body-correlation factor in fake news detection.

The “Getting Real about Fake News” dataset includes "Text and metadata from fake and biased news sources around the web". Since the dataset does not include "unbiased" or "true" news, the "Fake News Challenge" dataset was used to enrich its data.  Only the "*agree*”-labeled news data of this dataset was used to enrich the "Getting Real about Fake News" dataset. This work was done under the hypotheses that the label "agrees" in the "Fake News Challenge" dataset would be considered as "*True*".  Based on the fake news challenge “agree” is define as when "the body text agrees with the headline"

# **B. Exploratory Data Analysis**

To explore the data a value distribution analysis of the data was performed on the type column which consists of the label-classes.

****

As it is clear based on the above class distribution analysis. The classes in the dataset are not evenly distributed and many of the news are labeled as “bs”. The “agree” class that was added to the dataset in the data enrichment process holds the second rank in the class distribution matrix. While the distribution of classes is not ideal, however none of the classes were removed. The main motivation for this was to prevent the dataset from becoming a binary dataset, only including “bs” and “agree” news. Fakeness and truthfulness are not binary concepts and the datasets that are used for fake news detection should also not be binary, if one aims to perceive fake news detection as a real-world and non-abstract problem.

**C. Data Preprocessing**

After the data cleaning the following data preprocessing methods where applied on titles and text-bodies:

* Spell check and removing stop words
* Stemming
* Lemmatization
* Removing Punctuations

Using word cloud visualization, the following table represents how the preprocessing has changed the dynamics of the most frequent words in the dataset.

|  |  |
| --- | --- |
| **Word Cloud on titles before preprocessing** | **Word Cloud on titles after preprocessing** |
|  |  |
| **Word Cloud on text-bodies before preprocessing** | **Word Cloud on text-bodies after preprocessing** |
|  |  |

**Table - The impact of preprocessing represented using word cloud visualization**

1. **Distillation and Feature Engineering**

Beyond basic features such as bag of words (BOW) and TF-IDF, the below features were also extracted for titles and bodies.

* Doc2Vec Embedding
* Cosine similarity between title and body of every news
* LDA using Bow
* LDA using TF-IDF
* Sentiment
* Sentiment correlation between titles and bodies
* Cosine similarity between titles and bodies based on 10-dimensional LDA vectors

The aim here was to try to look at sub-areas and embedding (latent variables) within the factor topics, sentiments, and others so that additional features and parameters were extracted in order to help assessing the factor better. This would potentially provide the ability to get a higher accuracy score for the factor.

1. **Classifications**

The following classifiers were trained and evaluated on each of the features that were mentioned above and also on the combination of different features and the vectors that could be constructed based on the extracted features:

* Decision tree classifier
* Random forest classifier
* Naive Bayes classifier
* Neural network

The evaluation results of some of the conducted experiments are represented in the following table. More evaluation results and scores are available in the project notebook.

|  |  |
| --- | --- |
| **Naive Bayes on Title-Body Doc2Vec-Cosine-Similarity and Sentiment** | **Decision Tree on Title-Body Doc2Vec-Cosine-Similarity** |
| **Accuracy**: 0.7141961767471012 | **Accuracy**: 68.6305233469132 |
| **Naive Bayes on LDA-bow title-body-correlation and LDA-tf-idf title-body- correlation** | **Neural Network on LDA-bow title-body-correlation and LDA-tf-idf title-body-correlation** |
| **Accuracy**: 0.7141961767471012 | The result was not satisfactory |
| **Random Forest Classifier on all extracted features** | | |
| **Accuracy**: 0.7141961767471012 | | |
| **Visual representation of decision tree on title-body Do2Vec cosine similarity feature** | | |
|  | | |

1. **Testing the model on unseen document**

As a proof of concept, beyond constructing different vectors for describing the correlation between titles and body of a single document and evaluation of different classifiers to predict the type of news based on this factor, the cosine similarity between the title-body correlation of an iconic “agree” news and a randomly chosen (unseen) document was calculated. Later, based on the calculated cosine similarity, the label of the randomly chosen document was correctly predicted. This can be seen as an application of prototype theory of cognitive science in machine learning (fake-news detection). Similar to humans, here, an iconic exemplar (a prototype or stereotype) has been used for categorization of an unseen instance. See the attached notebook for more reflection and details on this topic.

# **3. Conclusion**

The problem of fake news is growing in a rapid pace as social media has become one of the major sources for generating and consuming news. People start believing that their perceptions about a particular topic are true without questioning the veracity of news. In this paper, we presented a systematic approach toward solving the problem of fake news detection. Various NLP and machine learning techniques were implemented in an attempt to identify the impact of each factor in detecting weather the news is truthful or fabricated. The goal was to define an equation consisting of multiple factors with different weightage to decide the fakeness-likelihood-score of an article. We demonstrate using the coefficients for each factor in the equation.

**Acknowledgment**

I would like to thank Professor Ali Arsanjani for giving me the opportunity to work on such important societal matter and for his invaluable guidance in preparation for this project.

**References**

Rubin, V. L., Chen, Y., & Conroy, N. J. (2015, November). Deception detection for news: three types of fakes. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community* (p. 83). American Society for Information Science.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22-36.

Thomas Davidson, Dana warmsley, Michael May and Ingma Weber. 2017 “Automated Hate speech detection and the problem of offensive language” Proceedings of the 11th international conference on Web and Social Media (ICWSM).

Wang, W. Y. (2017). " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

Thota, Aswini, et al. "Fake News Detection: A Deep Learning Approach." *SMU Data Science Review 1*.3 (2018): 10.

1. See https://en.wikipedia.org/w/index.php?title=Fake\_news&dir=prev&limit=500&action=history [↑](#footnote-ref-1)
2. https://www.kaggle.com/mrisdal/fake-news [↑](#footnote-ref-2)
3. https://github.com/FakeNewsChallenge/fnc-1 [↑](#footnote-ref-3)