Fake or True? Identify the Patterns behind Fake News

Zhiyu Zhang, Zihan Ling, Pizheng Zhang, Yafan Zeng, Xinyi Li
USC Marshall School of Business
December 2022

Table of Contents

1. Problem Identification

Problem Identification

1

Introduction

We take on the duty as a news platform to protect the reputation of our platform and try to increase readers subscriptions by providing authoritative informations and also help readers receive real news of what is happening in the world.

2

Our goal

Identify whether a news is fake or not using supervised machine learning model and increase the prediction accuracy.

3

Reason

1. Prevent fake news to get published on the website to preserve the reputation of the news platform.
2. Help the readers to have more access to real news.

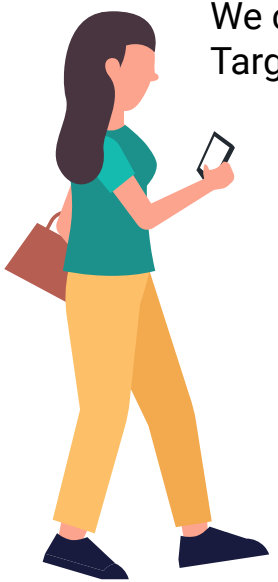
2. Exploratory Data Analysis

Data Source

We obtained our dataset from Kaggle.
There are two datasets: Fake.csv and True.csv with 44689 records and 4 columns originally.

- Fake News: 23481 (from '2015-03-31' to '2018-02-19')
- True News: 21417 (from '2016-01-13' to '2017-12-31')

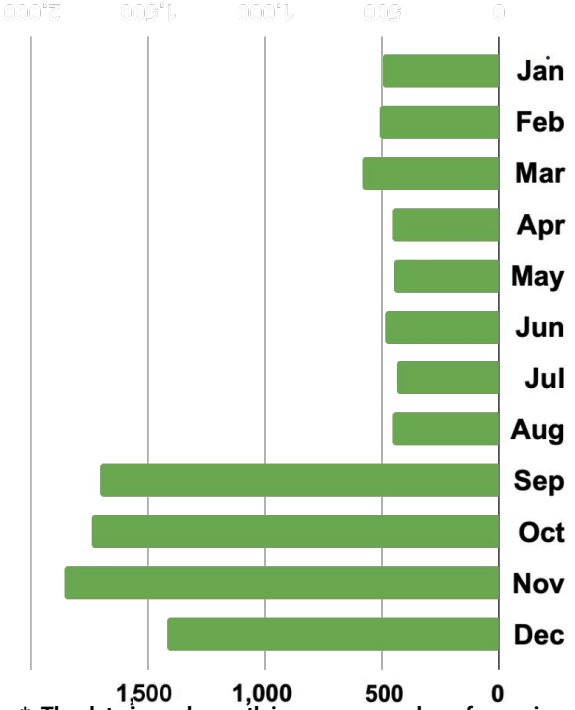
We created dummy variable: target as the fifth column to indicate if the record is fake or not.
Target = 1 means this news is fake.

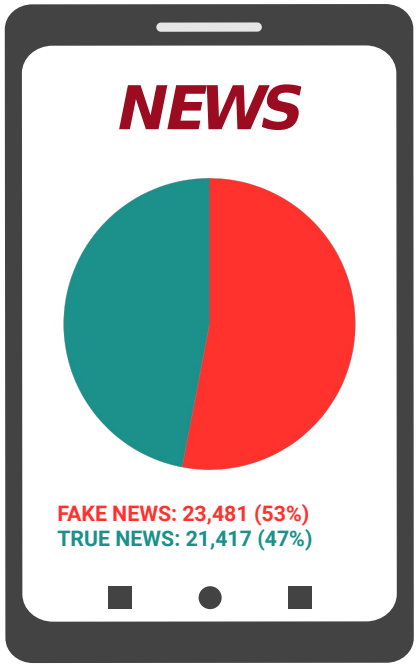


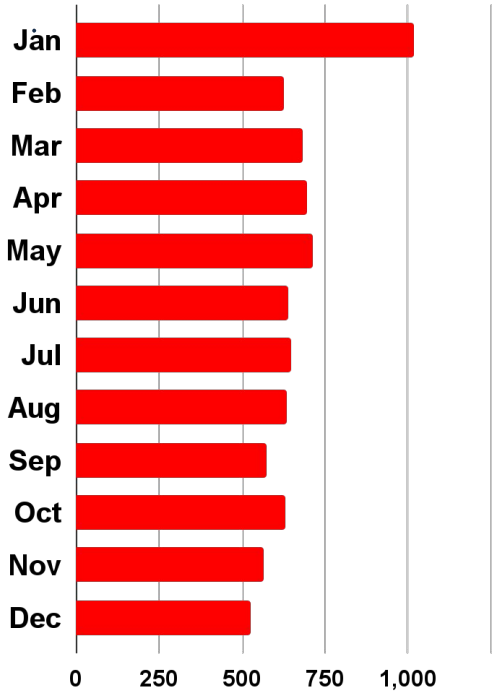
	title	text	subject	date	target
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	1
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	1
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	1
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	1
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	1

Source: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

Distribution of True and Fake News



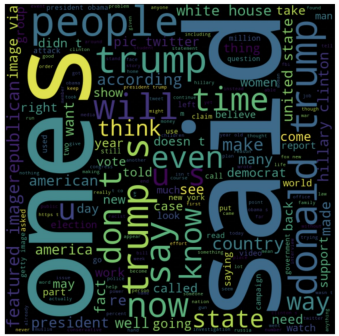


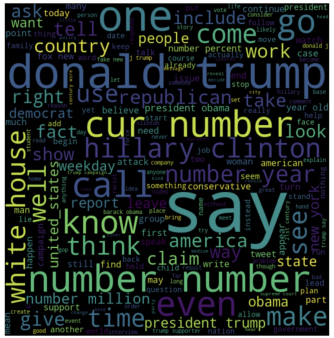


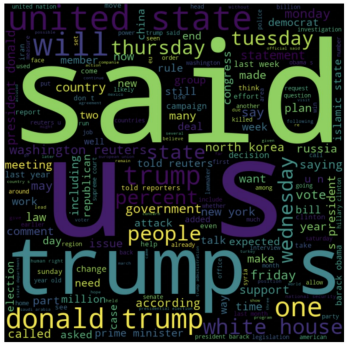
*: The data in each month is average number of news in each month.

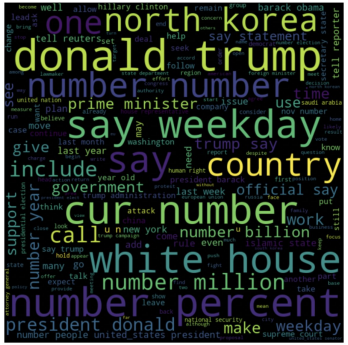
The number of true and fake news in different months is not evenly distributed. We can find that the number of true news from September to December is abnormally high. This will not have huge impact on our analysis but is an important point that needs to be focused on in the real world implications.

Word Clouds









Fake News (Before and After Data Preprocessing)

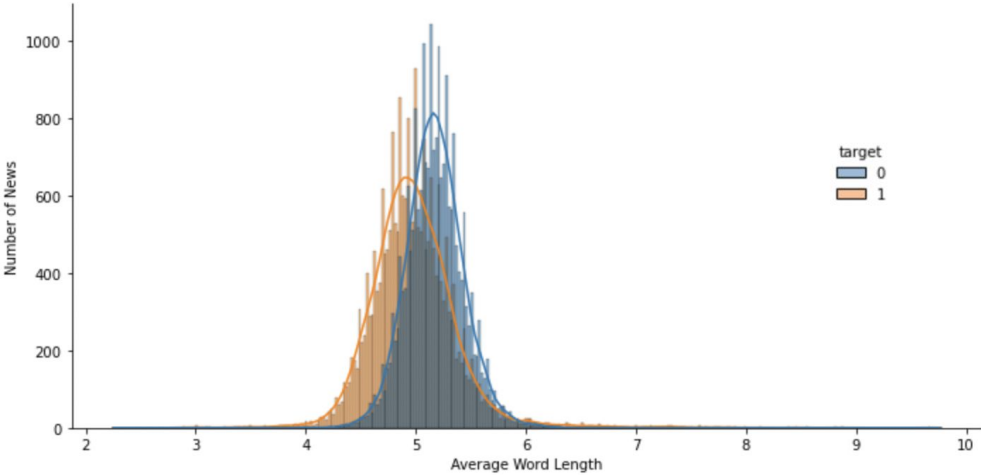
True News (Before and After Data Preprocessing)

From the word clouds, we can see that 2-3 words especially the word "said" account for a great proportion in both types of news before data preprocessing. These words cannot provide much information for our classification. However, after data preprocessing, top words are more balanced distributed. More meaningful words account for higher proportions.

Reference: https://www.geeksforgeeks.org/generating-word-cloud-python/#:~:text=For%20generating%20word%20cloud%20in,from%20UCI%20Machine%20Learning%20Repository

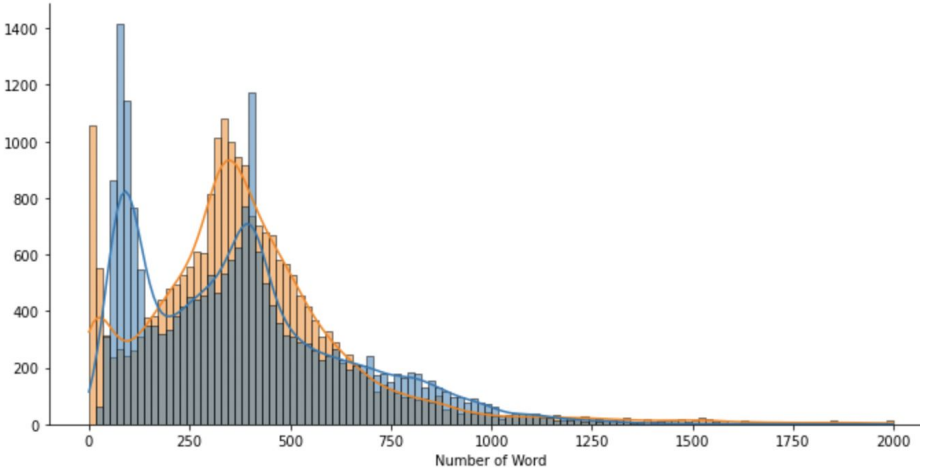
News Features

Average Word Length



Fake news has shorter average word length. Average average word length for fake news is less than 5 while that for true news is more than 5.

Number of Words



Most of fake news have 300-400 number of words while number of words for true news has two peaks (around 80 and 400 words).

3. Data Preprocessing

Data Preprocessing

We noticed that there are significant differences in data collections of true news and fake news, so we need to remove all these format differences to improve our model training process.

For example:

1. True news contains Reuters' information in the news text, while fake news does not.

	text
	WASHINGTON (Reuters) - The head of a conservat...
	WASHINGTON (Reuters) - Transgender people will...
	WASHINGTON (Reuters) - The special counsel inv...
	WASHINGTON (Reuters) - Trump campaign adviser ...

2. Fake news contains news type in the news title while true news does not.

	title
12558	WOW! TREY GOWDY On Harry Reid's Attack On Jim Comey: "I Didn't Know Mormons Used Drugs" [Video]
22578	BOILER ROOM – EP #55 – Roasting the Wretched Hive of Scum and Villainy
14187	"DEAD BROKE' HILLARY Ties Up Traffic In NYC For An Extravagant Appointment
4955	WATCH: Reliable Sources Explains 'Trumpbart' And How It Will Shape The Election (VIDEO)
14397	HOW REPUBLICAN LEADERS Are Willing To Sacrifice White House To STOP TRUMP

3. Other differences: date embedded in the news title; fake news text contains photo citations; etc.

Data Preprocessing - Part I

Remove Problematic Info

- Drop duplicates
- Remove news without dates
- Remove news with empty content

Format Cleaning

- Clean all format differences in the dataset using regex
- News Text
  - Remove reuters
  - Remove photo citations
- News Title
  - Remove news type
  - Remove news date embedded in the title

Other Regex Cleaning

- Date, weekday, country names, citation, twitter accounts
- Textacy: hashtags, numbers, currency_symbols, emojis, emails, quotation_marks, bullet_points, punctuation

Finally, we concatenate news title with news text because we believe that title and news content provide equivalent evidence on indicating whether the news is fake or not

Data Preprocessing - Part II

Stop Word Removal

- NLTK stopword package
- Regex: remove all the single length words because these words are most likely created by removing punctuation and have no meaning

Lemmatization

- We select lemmatization over stemming because it considers the word context and provides higher accuracy

Vectorization

- Count vectorization (uni-gram & bi-gram with max feature of 5000)
- TF-IDF vectorization (uni-gram & bi-gram with max feature of 5000)
- Word2vec

PCA

- We perform PCA to keep 80% of variance after count & TF-IDF vectorization due to the high dimensionality

4. Model Methodology

Modeling

Modeling Workflow 1

Brief introduction of the methods we use.

Model Details 2

Detailed explanation of the models we use.

Performance Evaluation 3

Evaluation of preprocessing techniques and models we use.

Comprehensive Analysis 4

Detailed summary of model evaluation results.

Part 1: Modeling Workflow

In this part, we fit both traditional machine learning models and cutting-edge deep learning models to make predictions.

Processed Data

We only pass cleaned data to traditional ML models

Vectorizing & Embedding

Countvectorizer, Tfidfvectorizer, and Word2Vec are used to construct input

Machine Learning Model

Naive Bayesian, Logistic Regression, Random Forest, SVC, CatBoost

Pro-/Unprocessed Data

We will try both cleaned and uncleaned data in DL models

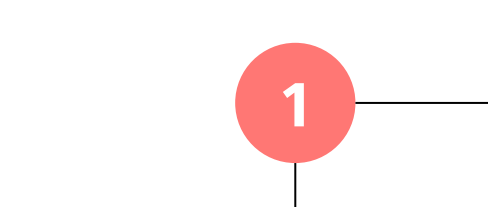
Embedding

Stanford “Glove.twitter.27B.100d” is used to construct input

Deep Learning Model

LSTM, BiLSTM with Attention, Transformer

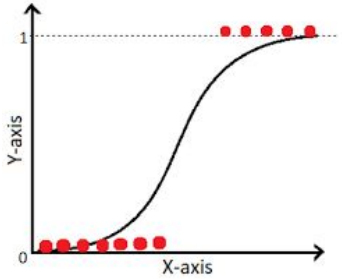
Part 2: Model Details



1

Logistic Regression

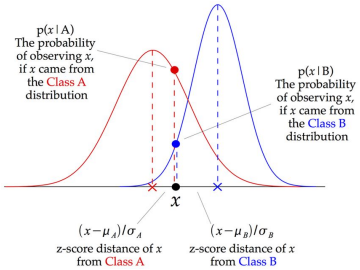
Simplest binary classification model with linear boundary



2

Naive Bayesian

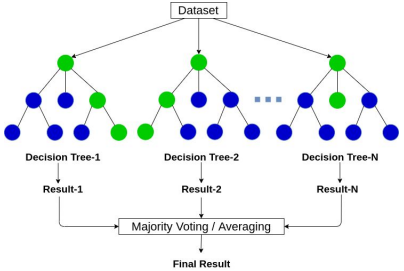
Traditional conditional probability model widely used in NLP



3

Random Forest

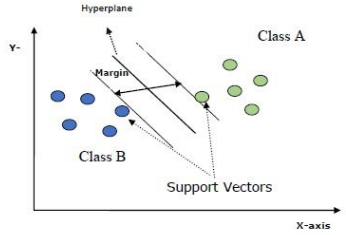
Ensembling model with multiple parallel decision trees



4

SVC

Find a best classification boundary in a higher dimensional space

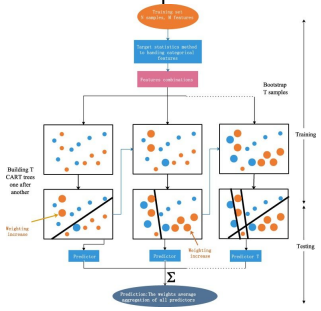


Part 2: Model Details

5

CatBoost

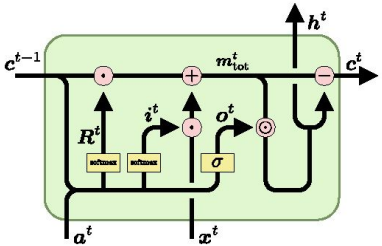
Novel gradient boosting model with permutation driven algorithm



6

LSTM

Sequential model with 1 cell and 3 gates that is widely used in NLP

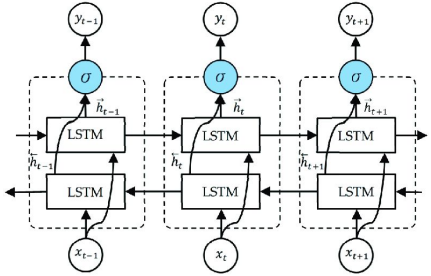


NEWS

7

BiLSTM

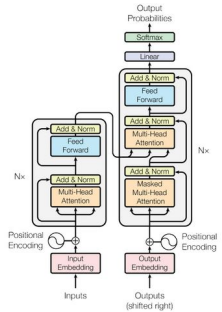
Combination of 2 LSTMs in order to increase the amount of information available



8

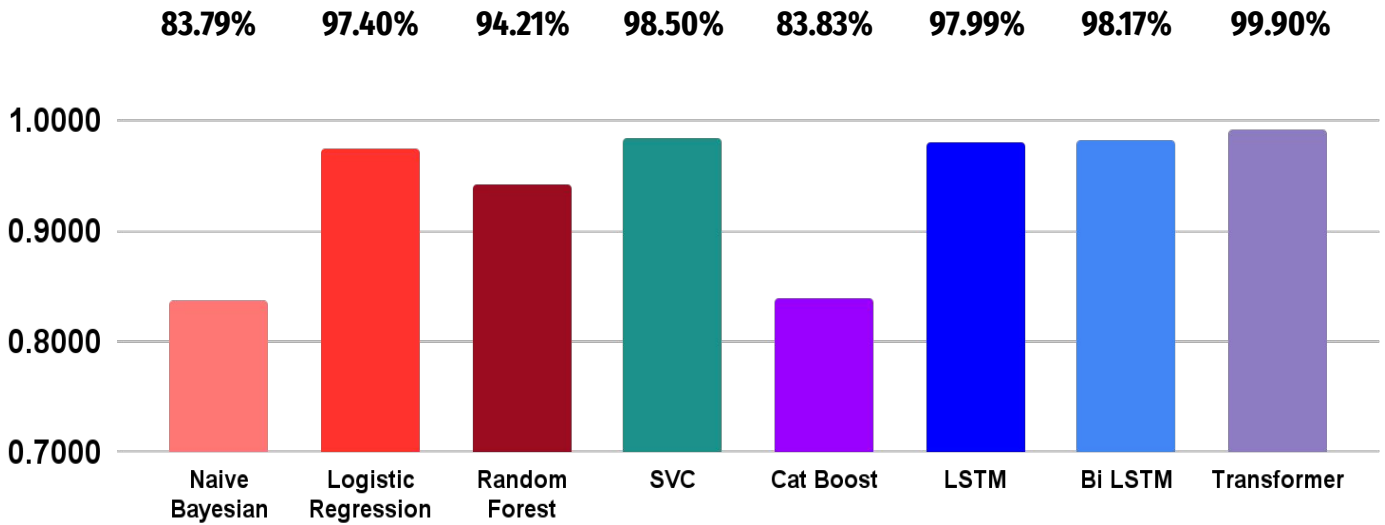
Transformer

New NLP model that can easily handle long-distance dependencies



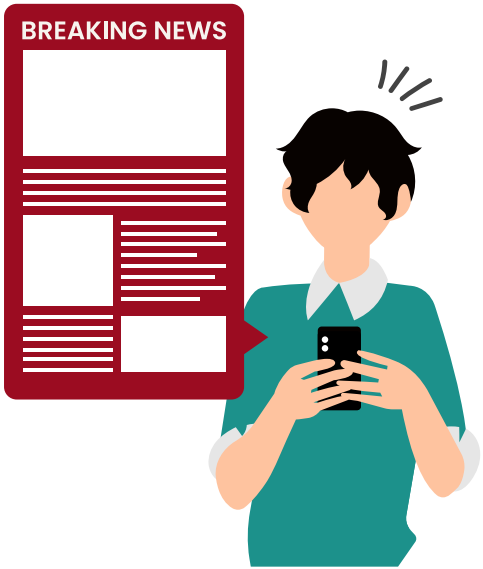
Part 3: Performance Evaluation - Model Performance

Model Performance Comparison (Testing Accuracy)



Except Naive Bayesian and Cat Boost models, all other models can provide reasonable predictions. Logistic Regression, SVC, and deep learning techniques tend to perform better than other techniques.

Part 3: Performance Evaluation - Data Processing Techniques 1

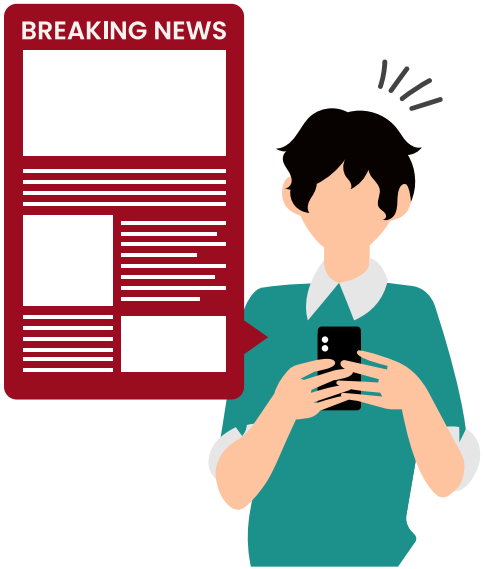


Processing Technique Comparison (Testing Accuracy)

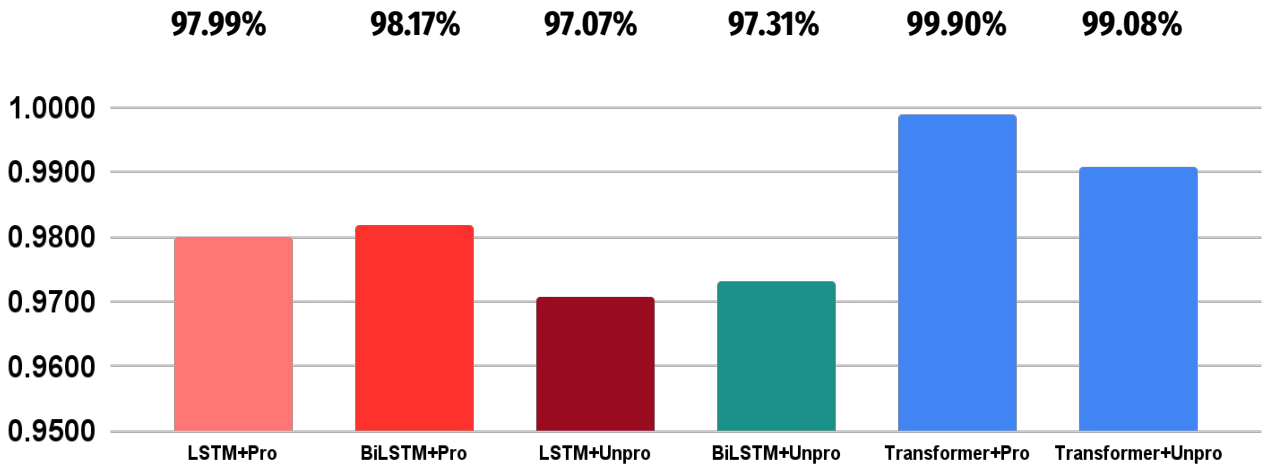
Legend: Count+Uni, Count+Bi, Tfidf+Uni, Tfidf+Bi, Word2Vec

Except random forest model, Tfidf vectorizer performs better than count vectorizer as well as word2vec model. As for unigram and bi-gram, there is not much difference.

Part 3: Performance Evaluation - Data Processing Techniques 2



Processing Technique Comparison (Testing Accuracy)



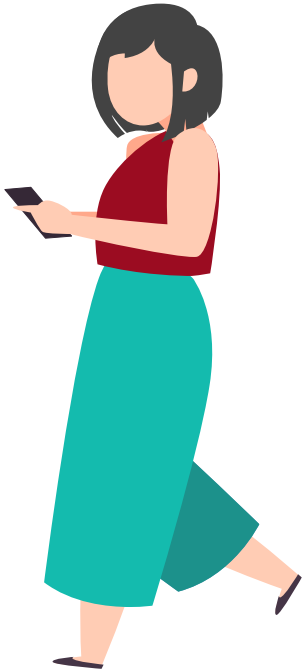
Prediction on processed data tends to be better than prediction on unprocessed data.
BiLSTM + Attention tends to perform better than LSTM in the classification task.

Part 4: Comprehensive Analysis - Summary

Comment on Pre-processing Techniques:

For traditional machine learning models, we mainly compare the performances of different vectorizers and number of grams. We can easily find that Tfidf-vectorizer performs normally better than Count-vectorizer. One possible reason is that some repeated words will occur in both true and fake news. It can have good prediction power only if we normalize it. Plus, for number of grams, different groups don't have obvious differences. It is because for most keywords in news text, single word and two consecutive words may have the exact same meaning (e.g. Trump and Donald Trump).

For deep learning models, prediction on processed data always performs better because true and fake news have structural difference in the original dataset. One possible reason is the method used in data scraping. That difference will be a serious noise in the prediction process.



Comment on Model Performances:

For all models we use, deep learning models perform better than traditional models on average because those techniques could capture the meaning of contexts. It is really hard for us to make judgement only considering the meaning of keywords in news.

For traditional machine learning models, logistic regression and SVC perform much better than others because the tasks we do is a binary classification task and these two models are designed and always perform best in binary problems.

5. Final Model Selection

Final Model: Bidirectional LSTM

Hyperparameter

Embedding:
- Input_length: *300*
- Trainable: *False*
- Weights = *embedding matrix**

Self-Attention Layer:
- Attention_activation: '*sigmoid*'

Compile:
- Optimizer: *'Adam'*
- Loss: *'binary_crossentropy'*
- Metric: *'accuracy'*

Bidirectional LSTM:
Units: *128*
Return_sequence: *True*
Recurrent_dropout: *0.25*
Dropout: *0.25*

Performance:
Accuracy: 0.9817
ROC-AUC: 0.9979

* Embedding matrix is from a pre-trained word vector from *stanford nlp* called *GloVe: Global Vectors for word Representation*: glove.twitter.27B.100d.txt

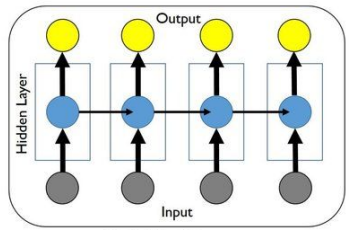
Bidirectional LSTM

Why we picked BiLSTM

- BiLSTM may read sentences forward and backward to leverage future context chunks to learn better representation of single words.
- BiLSTM is used to avoid vanishing and exploding gradient problem.
- BiLSTM model has high accuracy of **0.9817** and ROC-AUC of **0.9979**.

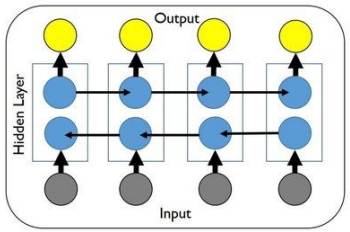
BiLSTM vs LSTM

- BiLSTM can produce a more meaningful output, combining LSTM layers from both directions.



LSTM Architecture

Hochreiter & Schmidhuber, 1997



BiLSTM Architecture

Graves & Schmidhuber, 2005

Future Improvement

Data Selection

Select data with consistent formats, so we can avoid potential overfitting problem.

1

Data Sufficiency

Collect more news data and diverse news topics to train the model.

2

Text Preprocessing

With allowance of time, we can do more regex and text preprocessing to discover issues with data formats to increase accuracy.

3

Modeling

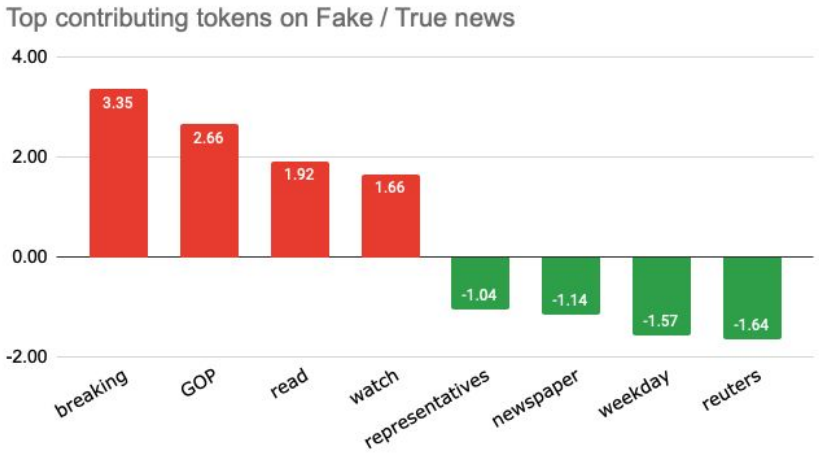
Adjust hyperparameter to further improve model accuracy.

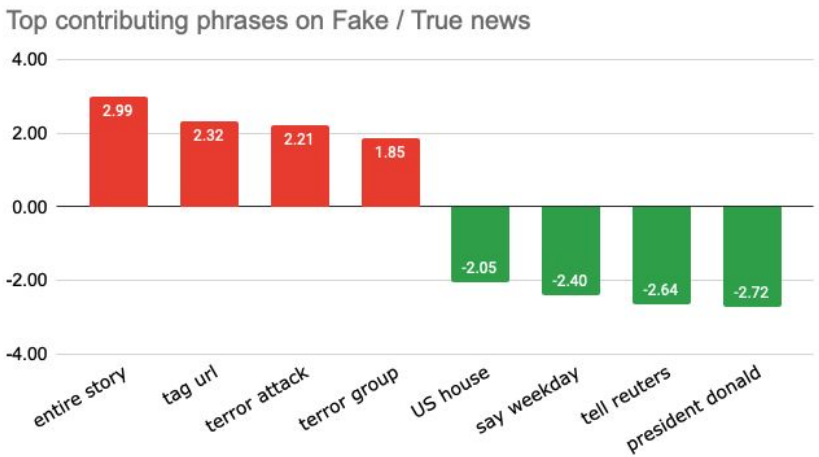
4

6. Business Implications

Which Words Contribute Most to Fake News?

By applying Logistic Regression models on preprocessed data with unigram and bigram count vectorization, we use the estimate coefficients of each token to measure its contribution on the target variable (fake news). Below are some results:





Note:

1. The tokens above is preprocessed by regex grouping and lemmatization. “url” and “weekday” represents a group of words.
2. The sign of number means whether it increases or decreases the probability of being fake news.
3. The magnitude of number measure the degree of contribution on the probability of being fake news.

Findings About Fake News Based On Token Contributions

Finding 1: Fake news tends to use the word "BREAKING" to create clickbaits.

In our datasets, 9 true news and 893 fake news contains "breaking" in titles

"**BREAKING**: Michael Flynn CRACKS – Will Testify To Mueller Against Trump Himself"

"**BREAKING**: The First Charges Have Been Filed In Mueller's Russia Investigation"

Finding 2: Fake news tends to partially expose stories and encourage to read more by providing external url.

"...In an exclusive interview with Breitbart News, the victim's father revealed that he had watched 30 seconds of the video of the horrific attack in that case. **For entire story: Breitbart News**"

"...If investigators find his campaign colluded with Russians, it's because so many people are so determined to bring him down.**Go HERE to read entire story**."

Finding 3: Fake news tends to cover terrific incidents to create panic and attract public attentions.

"...Fox News criticized Perry for her remarks following the deadly Manchester **terror attack** that left 22 people dead and 59 injured, including children..."

"...A joint intelligence bulletin from the FBI and the Department of Homeland Security found that the group we softly label white supremacists were responsible for 49 murders in 26 **terror attacks** over the last decade and a half…"

Findings About True News Based On Token Contributions

Finding 1: True news tends to come from large news agencies such as Reuters.

In our datasets, 4895 true news and 292 fake news contains 'Reuters' in contents.

"... Perhaps his arrogant comments are a result of a lack of (political) experience, Szydlo said in a statement **emailed to Reuters**..."

"...We tried to prevent the unrest in every possible way, but the protesters were totally out of control, Haryana Chief Minister Manohar Lal **told Reuters**..."

Finding 2: True news tend to provide the specific weekday in quotes.

"...a panel led by former U.N. chief Kofi Annan **said on Thursday**, adding that radicalization was a danger if problems were not addressed..."

"Iran and Saudi Arabia will exchange diplomatic visits soon, Tehran **said on Wednesday**..."

Finding 3: True news tend to call the presidents by their full names.

"President Barack Obama" appears in 2877 true news and 877 fake news.

"President Obama" appears in 145 true news and 2391 fake news.

"President Donald Trump" appears in 5749 true news and 913 fake news.

"President Trump" appears in 256 true news and 2568 fake news.

How Can Our Best Model Create Business Values?

Based on the statistics of the sample, we made following assumptions:

1. **News Volumes**
   100,000 news are created each year. 65% are true news, 35% are fake news.

2. **News Values**
   On average, a piece of true news creates $10,000 revenue; a piece of fake news leads to $5,000 loss.

3. **News Subscriptions**
   On average, a piece of true news brings 10 subscribers; a piece of fake news losses 15 subscribers.

Yearly Revenue Table

	Original Strategy	BiLSTM Strategy
Description	Publish with no censorship	Publish after BiLSTM Model screening
True News	65,000	63,927
Fake News	35,000	696
Subscription Gain	125,000	628,830
Total Revenue	$475 Million	$636 Million

Based on the assumptions, the BiLSTM strategy creates $635.79 million of revenue per year, overbeating the yearly revenue of original strategy by 34%. The BiLSTM strategy also brings a net increase of 628,830 subscribers, which is about 5 times as the original strategy.

Note:

1. In the dataset, 65% of the news in 2017 are true news and the other 35% are fake news.
2. The performance of BiLSTM is based on testing confusion matrix with 98.35% of recall and 1.99% of type I error.

7. Conclusion

Conclusion

In this project, we train 8 traditional and deep learning models to help the news platform detect fake news based on titles and texts.

Our best model is BiLSTM, which reaches the accuracy rate of 98.17% and ROC-AUC of 0.9979 in the test data. Based on this performance, our BiLSTM model can increase 34% of total revenue and 400% of subscribers comparing to the non-censorship strategy.

To improve our model performance in the real-word application, we need to collect more consistent data, perform more detailed data preprocessing, and further adjust the hyperparameters of our model.

8. Appendix

Appendix: Feature Size after Vectorization and PCA

Vectorizing/Embedding	Vectorization Feature Size	PCA (80%)	Final Feature Size
CountVectorizer + Uni-Gram	5000	Yes	670
CountVectorizer + Bi-Gram	5000	Yes	1171
TfidfVectorizer + Uni-Gram	5000	Yes	1820
TfidfVectorizer + Bi-Gram	5000	Yes	2246
Word2Vec	300	No	300

Appendix: Performance Summary 1

Model	Vectorizing/Embedding	Accuracy	ROC-AUC
Naive Bayesian	CountVectorizer + Uni-Gram	0.5170	0.5239
Naive Bayesian	CountVectorizer + Bi-Gram	0.6396	0.6349
Naive Bayesian	TfidfVectorizer + Uni-Gram	0.6509	0.6517
Naive Bayesian	TfidfVectorizer + Bi-Gram	0.7647	0.7651
Naive Bayesian	**Word2Vec**	**0.8379**	**0.8398**
Logistic Regression	CountVectorizer + Uni-Gram	0.9699	0.9700
Logistic Regression	CountVectorizer + Bi-Gram	0.9646	0.9644
Logistic Regression	**TfidfVectorizer + Uni-Gram**	**0.9740**	**0.9743**
Logistic Regression	TfidfVectorizer + Bi-Gram	0.9678	0.9679

Appendix: Performance Summary 2

Model	Vectorizing/Embedding	Accuracy	ROC-AUC
Logistic Regression	Word2Vec	0.9509	0.9509
Random Forest	**CountVectorizer + Uni-Gram**	**0.9421**	**0.9424**
Random Forest	CountVectorizer + Bi-Gram	0.9307	0.9308
Random Forest	TfidfVectorizer + Uni-Gram	0.9272	0.9276
Random Forest	TfidfVectorizer + Bi-Gram	0.9344	0.9343
Random Forest	Word2Vec	0.9307	0.9308
SVC	CountVectorizer + Uni-Gram	0.9753	0.9754
SVC	CountVectorizer + Bi-Gram	0.9626	0.9624
SVC	**TfidfVectorizer + Uni-Gram**	**0.9850**	**0.9852**

Appendix: Performance Summary 3

Model	Vectorizing/Embedding	Accuracy	ROC-AUC
SVC	TfidfVectorizer + Bi-Gram	0.9760	0.9760
SVC	Word2Vec	0.9521	0.9523
CatBoost	CountVectorizer + Uni-Gram	0.8258	0.8253
CatBoost	CountVectorizer + Bi-Gram	0.7986	0.7973
CatBoost	TfidfVectorizer + Uni-Gram	0.8080	0.8089
CatBoost	**TfidfVectorizer + Bi-Gram**	**0.8383**	**0.8383**
CatBoost	Word2Vec	0.8117	0.8117
LSTM	Unprocessed Data + GloVe	0.9707	0.9953
BiLSTM	Unprocessed Data + GloVe	0.9731	0.9960

Appendix: Performance Summary 4

Model	Vectorizing/Embedding	Accuracy	ROC-AUC
LSTM	Processed Data + GloVe	0.9799	0.9979
BiLSTM	Processed Data + GloVe	0.9817	0.9979
Transformer	Unprocessed Data + Bert-Base-Case + Glove	0.9908	0.9999
Transformer	Bert-Base-Case + Glove	0.9990	0.9999