Machine Generated Fake News Identifier Using Natural Language Processing

Saurab Siwakoti

Sandesh Koirala

University of Texas at Arlington
saurab.siwakoti@mavs.uta.edu sandesh.koirala@mavs.uta.edu

Abstract

Natural Language Processing is the computer representation, analysis, and generation of natural language texts. In this paper, we explore the use of this natural language processing techniques to the identify 'Fake News', that is misleading, and machine generated news stories published by a non-reputable source.

Using datasets from "Kaggle.com" which features an extensive human and machine generated news articles, we built a classifier that can make decisions about the provided information based on the content of the dataset with almost 95% accuracy.

Our model is developed on Python. It uses numpy, pandas, re and nltk libraries to extract, clean and filter the training data. The data extracted from Kaggle is first checked for null values and cleaned. After that, the processed dataset is then checked for duplicate entries. With all the null values and duplicate data cleared, the dataset is then transformed into two columns, label and text, removing id, subject and author columns. This data is again checked and filtered for stop words and punctuation marks. Computer does not understand English language for any calculation or evaluation, it needs to be converted into some sorts of machine-readable vectors. So, the processed text data is converted into vectors for classification. The vectorized data is then passed into different classifier models to see which one predicts the result with most accuracy. The model with the highest accuracy is then used to predict the result.

Introduction

This is an age of Artificial Intelligence. AI has progressed so much over the years; we can find its application on almost everything. These systems can tell what we are doing, can understand a whole book line by line or even predict what the stock market will be like in the future. Computers have started challenging human intelligence. Natural Language Processing is one such area of AI where computers are achieving intelligence every passing day. Currently, computers can process natural language and understand not only the syntactic but also semantic meaning of text pushing us to a new era of literature where computers help us create literature. But this also brings about its own curse of misuse. If misused, natural language processing can be used to cause social and political havoc. One such increasing misuse of natural language processing is to generate "fake news articles" to influence society. There are some systems available on the internet that can generate fake news, just by adding relevant words together. And, considering the horrible time that we are in, this false information could create chaos among us. NLP has several times been misused to defame politicians and people in high positions. Every large social media platform has seen and is battling misuse of NLP. This has brought forward a question of legitimacy of the any information on the internet.

As prospective computer scientists, this intrigued us. At the beginning of the semester, we started building a machine learning model that can identify "machine generated text" using natural language processing. This has been a tremendous learning opportunity for us. This report presents the achievements we have made during this period.

Related Work

There have been numerous attempts on creating a proper fake news identifying model in recent years. Some of these works have been used professionally whereas, others are projects from students and developers working throughout the globe.

One of the biggest tech giants of this era, Facebook has started a program, "Working to Stop Misinformation and False News." They are using third-party fact-checking organizations and different search algorithms to stop the spread of Fake News on its platform.

Digital Shadow, a cybersecurity startup is offering products that combat fake news. They specialize in removing these machine generated news from the portfolios and domains of the dark web.

Perimeterx, another cybersecurity company is tackling the problems of automated, pre-programmed bot distributed misinformation.

These are just a few commercial applications of fake news identification. The list goes on and on. Since, this is an uprising issue in the modern days of Internet, numerous ambitious companies are starting up, to give their part in taking down the bot generated misinformation and false news.

DarwinAI, a waterloo-based tech company has been working on developing a deep-learning based software that specializes on identifying fake news using stance detection.

Listed below are few more works that caught our attention:

1. Factmata – A startup based on fact-checking community.
2. Fabula AI – A company that uses geometric deep learning to detect fake news.

Problem Statement

To develop a machine learning model that identifies whether a given text is machine generated or human written using different classifying algorithms. The model will train using a corpus of data which consists of texts and labels to denote if its real or fake. These data will be filtered, tokenized and vectorized to transform them into data classifier algorithm's format. The vectorized data will then be fed into different classifiers to see which algorithm predicts the most accurate information. The best performing algorithm will then be used to predict if the provided text is real or fake.

Problem Solution

This project has been divided into different stages of development:

a. **Data Collection:**

A publicly available dataset from "Kaggle.com" featured an extensive text data containing total of 20800 "machine generated" and "human written" news articles with their corresponding "author" and "title". This dataset was decided to be used as the training and validation of the project.

The information about the acquired dataset follows:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   id      20800 non-null  int64
 1   title   20242 non-null  object
 2   author  18843 non-null  object
 3   text    20761 non-null  object
 4   label   20800 non-null  int64
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
```

Few rows from the top and bottom of the dataset is shown below:

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1
...	...	...	...	...	...
20795	20795	Rapper T.I.: Trump a 'Poster Child For White S...	Jerome Hudson	Rapper T. I. unloaded on black celebrities who...	0
20796	20796	N.F.L. Playoffs: Schedule, Matchups and Odds -...	Benjamin Hoffman	When the Green Bay Packers lost to the Washing...	0
20797	20797	Macy's Is Said to Receive Takeover Approach by...	Michael J. de la Merced and Rachel Abrams	The Macy's of today grew from the union of sev...	0
20798	20798	NATO, Russia To Hold Parallel Exercises In Bal...	Alex Ansary	NATO, Russia To Hold Parallel Exercises In Bal...	1
20799	20799	What Keeps the F-35 Alive	David Swanson	David Swanson is an author, activist, journa...	1

20800 rows × 5 columns

b. **Data Preprocessing:**

The efficiency of model depends largely on the quality of the data used. After the acquisition of data, several preprocessing was done. The resulting data-frame after each preprocessing is displayed below:

NULL value count over each column:

id	0
title	558
author	1957
text	39
label	0

NULL Value Count after deletion of rows with corresponding null values on column “title” and “text”:

id	0
title	0
author	1918
text	0
label	0

After the removal of null and duplicate rows, the dataset contains 19868 columns meaning 19868 news articles. Each article in the dataset was then randomly divided into training and validation dataset in the corresponding ratio of 0.75/0.25, respectively. This creates 14901 training article and 4967 articles for validation.

Then, the id, title, and author columns were removed as the results and decision is solely based on the text.

	label	title_text
0	1	House Dem Aide: We Didn’t Even See Comey’s Let...
1	0	FLYNN: Hillary Clinton, Big Woman on Campus - ...
2	1	Why the Truth Might Get You FiredWhy the Truth...
3	1	15 Civilians Killed In Single US Airstrike Hav...
4	1	Iranian woman jailed for fictional unpublished...
...	...	...
20795	0	Rapper T.I.: Trump a ’Poster Child For White S...
20796	0	N.F.L. Playoffs: Schedule, Matchups and Odds -...
20797	0	Macy’s Is Said to Receive Takeover Approach by...
20798	1	NATO, Russia To Hold Parallel Exercises In Bal...
20799	1	What Keeps the F-35 Alive David Swanson is an...

19868 rows × 2 columns

The text still had prepositions and punctuation marks and they had to be removed for precise training of the model. Thus, stop words were removed from the text, and this is the processed data that was ready to be tokenized and vectorized:

	label	title_text	clean_text
0	1	House Dem Aide: We Didn’t Even See Comey’s Let...	house dem aide didnt even see comeys letter ja...
1	0	FLYNN: Hillary Clinton, Big Woman on Campus - ...	flynn hillary clinton big woman campus breitba...
2	1	Why the Truth Might Get You FiredWhy the Truth...	truth might get firedwhy truth might get fired...
3	1	15 Civilians Killed In Single US Airstrike Hav...	15 civilian killed single u airstrike identifi...
4	1	Iranian woman jailed for fictional unpublished...	iranian woman jailed fictional unpublished sto...
...	...	...	...
20795	0	Rapper T.I.: Trump a ’Poster Child For White S...	rapper ti trump poster child white supremacyra...
20796	0	N.F.L. Playoffs: Schedule, Matchups and Odds -...	nfl playoff schedule matchup odds new york tim...
20797	0	Macy’s Is Said to Receive Takeover Approach by...	macys said receive takeover approach hudson ba...
20798	1	NATO, Russia To Hold Parallel Exercises In Bal...	nato russia hold parallel exercise balkansnato...
20799	1	What Keeps the F-35 Alive David Swanson is an...	keep f35 alive david swanson author activist j...

19868 rows × 3 columns

Machine learning models do not work with string data, they require an integer representation of the text data. So, both **CountVectorizer** and **TfidVectorizer**, from Sklearn were implemented in our filtered data. CountVectorizer creates a vector of frequency of each word appearing in the text. TfidVecorizer also creates a vector of frequency of the words but considering the weights of every word. TfidVectorizer gave out better results every time. Thus, it was used to vectorize our filtered data.

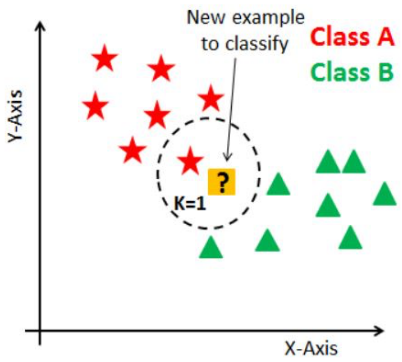
```
[ ] xfeed=df['clean_text']
    from sklearn.feature_extraction.text import CountVectorizer
    from sklearn.feature_extraction.text import TfidfVectorizer
    cv= TfidfVectorizer()
    x=cv.fit_transform(xfeed)
```

c. **Model Creation:**

The filtered data was first divided into training data and test data. Classification was done using KNN classifier, Decision tree and Support Vector Machine. KNN classified the data with 83% accuracy, Decision tree classified the data with 92% accuracy and Support Vector Machine did it with 96% accuracy.

KNN Classifier

K-Nearest Neighbors uses data and classifies new data points based on similarity measures. It works by finding distances between a query and all the examples in the data, selecting the specified number examples closest to the query, then voting for the most frequent label.



Our model implements KNN using KNeighborsClassifier from Sklearn library. This classifier resulted in 83% accuracy.

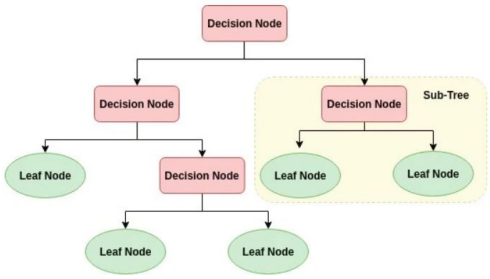
```
knn = KNeighborsClassifier(n_neighbors=4)
knn.fit(train_feature, train_class)
print("Test set predictions:\n{}".format(knn.predict(test_feature)))
print("Test set accuracy: {:.2f}".format(knn.score(test_feature, test_clas

Test set predictions:
[0 1 0 ... 0 0 0]
Test set accuracy: 0.83
```

Decision Tree

Decision Tree classifier is a type of supervised machine learning algorithm where the data is continuously split according to a certain parameter. The tree consists of:

- Nodes: Test for the value of a certain attribute.
- Edges/ Branch: Outcome of a test and a connection to the next node or leaf.
- Leaf nodes: Terminal nodes that predict the outcome.



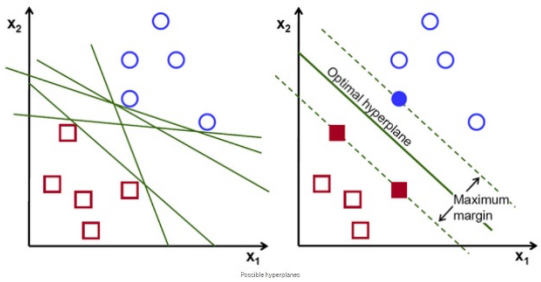
Our model implemented Decision trees using DecisionTreeClassifier from sklearn library. After fitting the vectorized data and classifying, the model predicted the output with an accuracy of 92%.

```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier()
tree.fit(train_feature, train_class)
print("Training set score: {:.3f}".format(tree.score(train_feature, train_class)))
print("Test set score: {:.3f}".format(tree.score(test_feature, test_class)))

Training set score: 1.000
Test set score: 0.922
```

Support Vector Machine

Support Vector Machine (SVM) performs classification based on a plot. Each data is first plotted as a point in n-dimensional space with the value of each feature being the value of a particular coordinate and classification is then performed by finding the hyper-plane that differentiates the two classes very well.



Our model used linearsvc from sklearn for support vector machine algorithm. It predicted the results with 96% accuracy.

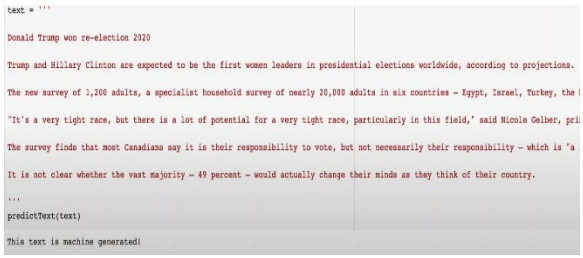
```
linearsvm = LinearSVC().fit(train_feature, train_class)
print("Test set score: {:.3f}".format(linearsvm.score(test_feature, test_class)))

Test set score: 0.961
```

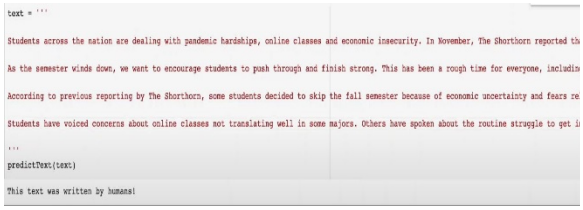
Out of all the classification models implemented, Support Vector Machine predicted the results with highest accuracy. Thus, this model was chosen for classification.

d. **Prediction:**

The Parameter were tuned for optimum accuracy. And hence, a model was created that can identify if a given set of texts is either machine generated or human written. After all the final tweaks and commits, our model takes a set of texts, vectorizes the data, compares it with our dataset, and displays if the text is machine generated or human written.



In the image above, a random text was generated from fake news generator. This text was passed on to our model. It was checked and compared with the data our model was trained on, classified using Simple Vector Machine and the model predicted the output, "**This text is machine generated!**".



Here is another trial with an actual human written text. A portion of an article was copied from "The ShortHorns" newspaper and was fed into our model. Our model checked the text with the trained data, classified it and predicted the output, "**This text was written by humans!**".

On performing several tests, our model was able to successfully predict results.

Conclusion

This project shows prominent possibilities for the use of Machine Learning models to identify misinformation as such. This project can be further continued to provide a better User Interface. Another possibility could be to create wrappers for the library in the form of browser extensions. These extensions could be accessed from browsers to feed news link in order to train the model or to warn visitors of certain web link that a certain news site could be misinformation. All in all, this has been a very successful project. This project strives to bring into light the current widespread situation of misinformation and also brings into light how, as computer scientists, we can fight misinformation at the root of its cause.

References

Ahmad, I., Yousaf, M., Yousaf, S., & Ahmad, M. (2020, October 17). Fake News Detection Using Machine Learning Ensemble Methods. Retrieved December 14, 2020, from https://www.hindawi.com/journals/complexity/2020/8885861/

Detecting Fake News with Scikit-Learn. (n.d.). Retrieved December 14, 2020, from https://www.datacamp.com/community/tutorials/scikit-learn-fake-news

Dounis, F. (2020, June 05). Detecting Fake News With Python And Machine Learning. Retrieved December 14, 2020, from https://medium.com/swlh/detecting-fake-news-with-python-and-machine-learning-f78421d29a06

Edell, A. (2018, January 17). I trained fake news detection AI with 95% accuracy, and almost went crazy. Retrieved December 14, 2020, from https://towardsdatascience.com/i-trained-fake-news-detection-ai-with-95-accuracy-and-almost-went-crazy-d10589aa57c

The Fake News Detector. (n.d.). Retrieved December 14, 2020, from https://www.albany.edu/ceas/news/fake-news-detector

Mohd Sanad Zaki RizviA computer science graduate. (2019, December 16). An Exhaustive Guide to Detecting Neural Fake News using NLP. Retrieved December 14, 2020, from https://www.analyticsvidhya.com/blog/2019/12/detect-fight-neural-fake-news-nlp/

Zellers, R. (n.d.). Defending Against Neural Fake News. Retrieved December 13, 2020, from https://proceedings.neurips.cc/paper/2019/file/3e9f0fc9b2f89e043bc6233994dfcf76-Paper.pdf