**[Group 1]**
Jack Charbonneau
Jannik Haas
Jian Liu
Mario Arduz
Joan Wong

# Creating Models to Discriminate Between Authentic and AI-Generated Tweets

## 1 Introduction

In recent years, there has been great interest in the field of Natural Language Processing (NLP). NLP is a broad term for many tasks associated with the computation of human language, such as speech recognition, translation, document summarization, and text generation. A lot of buzz was created in the area of text generation following OpenAI's release of the GPT-2 model in 2019. The model was trained with 40GB of text data scraped from various websites across the Internet. Given all of the words in a sequence of text, the model was tasked with predicting the next word. The model was extremely successful, and its ability to continually predict the "next word" allowed it to string together long sequences of text that resemble something that could be made by a human. The ability to create extremely realistic text that mimics the style of human language opens up a variety of ethical concerns. In fact, the creators of GPT-2 were so concerned that the model would be used for malicious purposes that they initially refused to release the full size model, instead opting to start by releasing a small version of the model and monitoring its use to determine if a larger model should be released [A].

A large part of the ethical concerns surrounding AI-generated text has to do with the ability (or inability) to take a sample of text and determine if it really belongs to the human author that it is attributed to. In this project, we will endeavor to implement and describe a few AI-based techniques for classifying text as real (human-written) or fake (AI-generated). In particular, we will be using GPT-2 to generate a data set of fake tweet messages from a variety of authors. We will then combine this data set of fake tweet messages with a data set of real tweet messages and attempt to train models that can take the text of a tweet and determine if it was written by the real author or generated by GPT-2. We will conclude the project by evaluating the success of the classification models, presenting the results, and discussing how the results could be used to create generated text that is even harder to detect.

# 2 Methodology

## 2.1 Data Collection

The team ultimately decided to conduct experiments with data from two Twitter users. The first chosen user is @realDonaldTrump. President Trump is well known for his activeness on Twitter and his unique manner of speech. Most of Trump's tweet messages are related to politics, and some may be written or curated by his staff members. The other user whose data was used in this project is @chrissyteigen. Chrissy is a celebrity who has 12.7 million Twitter followers and is very active on the platform. Teigen seems to be largely free to tweet about anything she wants, and her tweets are often centered around humor rather than politics. We believe that data from these two users will provide a way to compare how results may differ when an AI is mimicking someone whose content is primarily focused on one serious subject (politics) versus someone who mentions any number of frivolous topics that are currently piquing their interest.

### 2.1.1 Real Tweets

Real tweet messages were collected using the `GetOldTweets3` Python library. The official Twitter API limits the user to scraping tweet messages from the last seven days. `GetOldTweets3` allows us to bypass this limitation and enables the use of larger data sets. The library is very simple to use and allows the collection of many tweet messages using only a few lines of code. The tweets that were collected include stand alone tweets and tweets that are replies to other tweets. Five thousand tweets were collected for each user. The text of the tweets was exported to CSV files, while the other data (date, time, number of favorites, etc.) was discarded. Samples of real tweets from Trump and Teigen can be found in Appendices 1 and 2, respectively.
*Note: Some of the collected tweets are empty-quoted tweets with no text which were discarded, making the actual number of tweets per user less than 5,000.*

### 2.1.2 Fake Tweets

The GPT-2 model was utilized through the `gpt-2-simple` Python library. This model provides a simple interface for downloading GPT-2 and fine-tuning it on a data set of your choice. For each user of interest, we fine-tuned the model for 100 steps using a data set of 1,000 real tweet messages. Once the model was trained, we generated 5,000 fake tweets for each user. GPT-2 is generally geared toward generating long-form text and is not very good at smoothly terminating text at a certain length. Because twitter messages are limited to 280 characters, we truncated the length of the fake tweet messages generated by the GPT-2 model to be within 280

characters. Generated tweet messages were then exported to a CSV file. Samples of fake tweets from Trump and Teigen can be found in Appendices 3 and 4, respectively.
*Note: On some occasions, GPT-2 seems to have hiccups resulting in random start and end of generation tags being placed in the middle of tweets. These broken samples are removed, making the actual number of fake tweets per user less than 5,000.*

## 2.2 Discriminative Models

The team created two models to attempt to classify tweet messages as real or fake. The first model used was a Random Forest (RF). The primary benefit of this model is its interpretability. RFs are fairly easy for humans to understand and can provide a lot of information about which features are most useful in identifying fake text. The downside to the RF for this application is that human language is very complex and nuanced, and it can be difficult to reduce text to a number of simple features without losing a lot of information.

The second model utilized is a Convolutional Neural Network (CNN). Unlike a RF, a CNN is not very easy for humans to understand and does not provide a lot of information about how it is able to classify text effectively. The advantage of a CNN is that, rather than reducing text to simple features, it can practically take the entire text as input during its classification.

### 2.2.1 Random Forest

Random Forest (RF) fits multiple decision trees with random subsets of the training set, and the model's prediction comes from the majority's vote (or average) of the decision trees. The idea of RF is that *many* uncorrelated trees can outperform (or reduce errors made by) any one of them alone.

We will use the RF Classifier provided by the `sklearn.ensemble` Python package. The independent variables (X) are features extracted from the training input sample of tweet messages, and the dependent variable (y) is the target value, which is the author label in classification. We used all default parameters for our RF Classifier, except we set the number of trees in the forest to be 100 (n_estimators = 100).

The features extracted from the tweet messages attempt to capture writing styles, characteristics, preferences, and patterns of authors. As listed below, there are a total of 16 features extracted for the purpose of this project; some of which were identified and inspired by previous works [B, C].

#### Features

*Maybe the author likes to refer to someone/something?*

- **NUM_HASHSTAG**
  The number of hashtags in each tweet.
- **NUM_MENTIONS**

The number of @mentions in each tweet.

*Maybe the author tends to be very pessimistic or optimistic?*

- **POSITIVE_SENTIMENT**
  Proportion of text that falls in the positive category * 100
- **NEUTRAL_SENTIMENT**
  Proportion of text falls in the neutral category * 100
- **NEGATIVE_SENTIMENT**
  Proportion of text falls in the negative category * 100
- **COMPOUND_SENTIMENT**
  Composite sentiment score * 100

*Note: The sentiment scores are computed by VADER sentiment analysis tool from* `vaderSentiment.vaderSentiment` *for the purpose of identifying or detecting emotions within social media text.*

*Maybe the author likes to write in complete sentences?*

- **NUM_SENTENCES**
  The number of sentences in each tweet.

*Note: The following features are computed after removing all hashtags and mentions, and tokenizing the tweet message using the* `nltk.tokenize` *Python library. Additionally, stop words are mostly excluded in the computation, since they are very commonly-used and unimportant in the tweet message. Stop words are acquired from* `nltk.corpus`.

*Maybe the author likes to be long-winded? Use big words?*

- **WORD_COUNT**
  The number of alpha-numeric tokens, excluding stop words, in each tweet.
- **AVG_WORD_LENGTH**
  The average length of alpha-numeric tokens, excluding stop words, in each tweet.
  OR
  The average length of alpha-numeric tokens in each tweet if it contains only stop words.
  OR
  Can be 0 otherwise
- **AVG_SENTENTIAL_LENGTH**
  WORD_COUNT / NUM_SENTENCES if NUM_SENTENCES > 0
  OR
  Can be 0 otherwise

*Maybe the author likes to use a lot of adjectives or verbs to express thoughts/feelings?*

- **NNP** (Proper noun, singular)
  (Number of tagged tokens / WORD_COUNT) * 100
- **NN** (Noun, singular or mass)
  (Number of tagged tokens / WORD_COUNT) * 100
- **JJ** (Adjective)
  (Number of tagged tokens / WORD_COUNT) * 100
- **NNS** (Noun, plural)
  (Number of tagged tokens / WORD_COUNT) * 100
- **RB** (Adverb)
  (Number of tagged tokens / WORD_COUNT) * 100

*Note: There are many parts of speech tags [D], but we only chose the top 5 most-frequently used in the corpus (across all data sets) identified by `nltk`.*

*Maybe the author likes to use symbols like & or / instead of text?*

- **NONWORD_CHARS**
  (Number nonalpha-numeric tokens / number of characters in tweet message) * 100

### 2.2.2 Convolution Neural Network

Convolutional Neural Networks (CNNs) are well known for their use in image classification tasks, but they can also be used for text classification. The model structure used in this project is partially based upon the "1D CNN for text classification" example in the `Keras` documentation but also takes influence from other sources [E, F, G]. The model was implemented in Python using the `Keras` neural network library. The structure of the model along with the shape of its inputs and outputs is summarized in the table below.

| Layer | Input Shape | Output Shape |
|---|---|---|
| Input | (None, 140) | (None, 1400) |
| Embedding | (None, 140) | (None, 140, 50) |
| Conv1D | (None, 140, 50) | (None, 137, 32) |
| GlobalMaxPooling1D | (None, 137, 32) | (None, 32) |
| Dense | (None, 32) | (None, 10) |
| Dense | (None, 10) | (None, 1) |

**Table 1.** Structure of CNN

When using the CNN, we do not create features in the same way that we did when creating a Random Forest (RF). The text of each tweet simply has stop words removed from it and is then tokenized into a sequence of word bigrams. Each sequence has a length of 140, which corresponds to the maximum length of a tweet. The sequence of bigrams is then used as input to the model. The first layer of the model is an embedding layer which restructures the input into a more usable format. The output of the embedding is passed to the one-dimensional convolutional layer, which learns features from the sequences. The convolutional layer uses a kernel (or window) size of 4. A pooling layer is then used to reduce dimensionality before passing on to two dense layers that ultimately output a single classification prediction using the sigmoid activation function. Between each layer in the network, a dropout layer with a rate of 0.5 was added in an attempt to help combat overfitting.

# 3 Experiments

After carrying out the data collection processes described in our methodology, we had a total of four data sets available with which to conduct experiments.
[Trump] Real tweet messages (with 4,659 samples),
[Trump] Fake tweet messages (with 3,428 samples),
[Teigen] Real tweet messages (with 4,861 samples), and
[Teigen] Fake tweet messages (with 4,600 samples).

We performed four experiments, in which we trained a Random Forest (RF) model and a Convolution Neural Network (CNN) model with different combinations of tweet messages to predict authors of tweet messages. The combined data sets are:

1. [Trump] Real tweets and [Trump] Fake tweets
2. [Teigen] Real tweets and [Teigen] Fake tweets
3. [Trump] Real Tweets and [Teigen] Real Tweets
4. [Trump] Fake Tweets and [Teigen] Fake Tweets

For each combined data set, 70% of the observations were used as the training set, and 30% were used as the testing set. The results of the experiments were used to evaluate the performance of the GPT-2 model. Experiments 1 and 2 were used to determine whether our models were able to distinguish between real and fake tweets, which indirectly evaluates the performance of the GPT-2 model. Experiments 3 and 4 were used to evaluate how well our models can distinguish between different authors of tweet messages, which also indirectly evaluates how well the GPT-2 model does in capturing unique writing distinctions between different authors.

# 4 Results

The four experiments described in the previous section were carried out to evaluate the performance of the models. Each experiment involves classification between two classes. The data and classes being tested are listed in the “Combined Data Sets” column of the tables, and the results are recorded in the remaining columns.

| Combined Data Sets | Model Accuracy | Top 5 Feature Importances |
|---|---|---|
| Experiment 1<br><br>[Trump]Real<br>[Trump]Fake | Accuracy: 0.70 | WORD_COUNT<br>NONWORD_CHARS<br>AVG_WORD_LENGTH<br>COMPOUND_SENTIMENT<br>NEUTRAL_SENTIMENT |
| Experiment 2<br><br>[Teigen]Real<br>[Teigen]Fake | Accuracy: 0.59 | AVG_WORD_LENGTH<br>NONWORD_CHARS<br>COMPOUND_SENTIMENT<br>NEUTRAL_SENTIMENT<br>WORD_COUNT |
| Experiment 3<br><br>[Trump]Real<br>[Teigen]Real | Accuracy: 0.91 | NNP<br>AVG_WORD_LENGTH<br>NONWORD_CHARS<br>WORD_COUNT<br>NN |
| Experiment 4<br><br>[Trump]Fake<br>[Teigen]Fake | Accuracy: 0.89 | NNP<br>AVG_WORD_LENGTH<br>NONWORD_CHARS<br>WORD_COUNT<br>COMPOUND_SENTIMENT |

**Table 2.** RF Performance

| **Combined Data Sets** | **Model Accuracy** |
|---|---|
| Experiment 1<br>[Trump]Real<br>[Trump]Fake | Accuracy: 0.77 |
| Experiment 2<br>[Teigen]Real<br>[Teigen]Fake | Accuracy: 0.63 |
| Experiment 3<br>[Trump]Real<br>[Teigen]Real | Accuracy: 0.94 |
| Experiment 4<br>[Trump]Fake<br>[Teigen]Fake | Accuracy: 0.91 |

**Table 3.** CNN Performance

# 5 Discussion

In each of the experiments we carried out, both models were able to obtain an accuracy that is significantly better than random guessing. The models had much better performance when classifying between two different users (Experiments 3 and 4) as opposed to when they were classifying between real and fake versions of the same user (Experiments 1 and 2). Thus, we can see that GPT-2 is capturing unique features in writings of different authors and generating highly personalized texts. However, the fact that our models were able to identify fake tweet messages with considerable accuracy indicates that the GPT-2 is by no means perfectly suited for creating deceptive tweets. Both of the models that we employed had a higher accuracy spotting fake Trump tweets than spotting fake Teigen tweets. This seems to suggest that certain manners of speaking are harder to classify as real or fake. In this particular case, it seems that the brevity of Teigen's tweets make them hard to classify. If the text is only a few words long, it is hard to draw any meaningful conclusions.

## 5.1 Random Forest

Looking specifically at the results of the Random Forest (RF) model, we can see that it is possible to obtain a respectable classification accuracy by extracting features from text. In addition to the RF being useful as a classification tool, it also provides a lot of insight into what features are actually useful in the classification. We examined feature importances of the RF model in all four experiments and were able to identify several important features in detecting fake tweet messages and authors.

The useful features in *distinguishing fake and real tweet messages* (overlapping features in Experiments 1 and 2) are:

- WORD_COUNT
- NONWORD_CHARS
- AVG_WORD_LENGTH
- COMPOUND_SENTIMENT
- NEUTRAL_SENTIMENT

Word count is one of the most important features in generating fake text. The GPT-2 model is made for long-form text generation, and it is currently not capable of crafting text of a certain size. For that reason, many of the generated tweet messages had to be truncated in order to comply with Twitter's 280 character limit. Finding a way to force generated text to be within a certain length would definitely help create more believable tweets. Sentiment features are also found to be significant in classifying tweet messages. These features allow the model to capture or gain insight to the author's opinion or attitude towards certain topics. We suspect that the reason why neutral sentiments weighed more than the other types of sentiments in both experiments was because the generated texts from the GPT-2 model may be more generic than what the actual authors wrote.

The useful features in *distinguishing the authors of tweet messages* (overlapping features in Experiments 3 and 4) are:

- NNP
- AVG_WORD_LENGTH
- NONWORD_CHARS
- WORD_COUNT

The features that are most influential in classifying the author of a tweet message seem to be centered around the author's style of writing. The use of proper nouns is the most important feature, which seems reasonable, since many of President Trump's tweet messages make named references to other political leaders. On the other hand, Teigen does not seem to reference other people as much. Word length, word count, and the use of nonword characters are other important features, which are also related to the author's style of writing. For example, Trump writes relatively long tweet messages, while Teigen prefers to keep her tweet messages short. Additionally, Teigen also makes occasional use of excess punctuation and ASCII art as shown in Appendix 2.

## 5.2 Convolutional Neural Network

Although the CNN was able to achieve good accuracy, it seems to have a problem with overfitting. The plot of training and testing accuracies shown below demonstrates that the training accuracy of the model is extremely high.

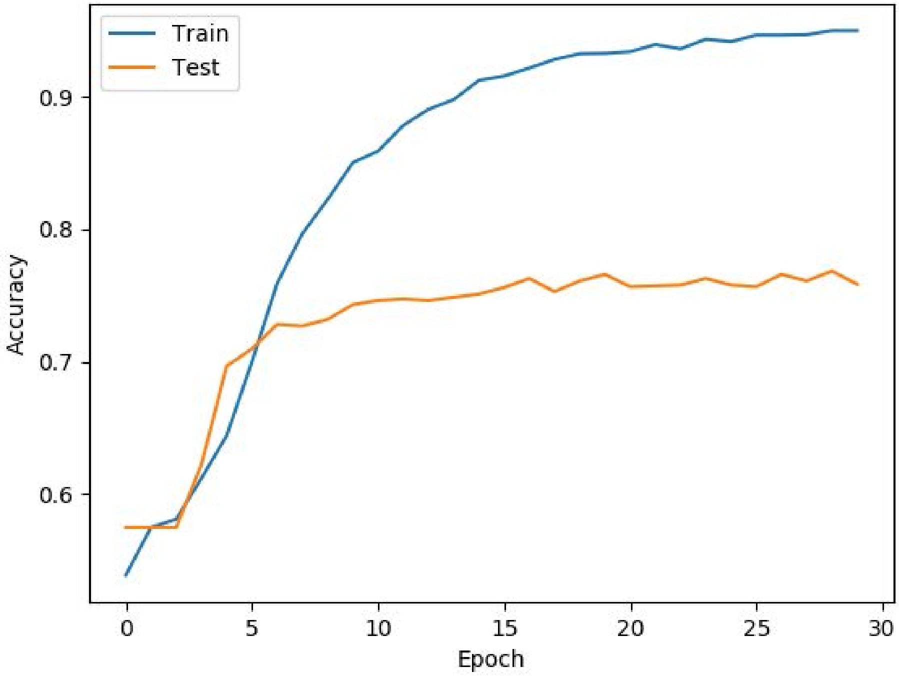

**Figure 1.** CNN Performance on Training and Testing Data Set

In the first few epochs, the training and testing accuracies remain quite comparable to one another. However, as the number of epochs increases, the training accuracy continues to skyrocket, while the rate at which the test accuracy increases begins to slow down. Many strategies were employed to try to reduce this overfitting. Increased dropout rates, reduced learning rates, reduced model complexity, and varying batch sizes were all unsuccessful in reducing this overfitting. We would have liked to see if using a larger data set could help, but even `GetOldTweets3` seems to have limitations to how much data it can fetch. Although the model obtained decent results, it seems reasonable to assume that these results could be improved if we had been able to find an effective way to address this issue.

# 6 Conclusion

This project demonstrates the successful implementation of two models for identifying AI-generated tweets. Our results showed that the Convolutional Neural Network (CNN) model outperformed the Random Forest (RF) model, which highlights the importance for understanding the strengths and drawbacks of using a highly performant model or a highly interpretable one. The use of the RF model allowed us to discover some useful features that may help in determining if a message is real or fake. In addition to finding the useful features, we outlined why these features may be important and how knowledge of these features could be used to create more deceptive generators. With the implementation of the CNN, we showed that although deep learning can be used to create fake text, it can also be used to detect it. The CNN achieved considerable accuracy and could certainly be improved upon by a more experienced group of researchers in the future.

# References

[A]https://openai.com/blog/better-language-models/
[B]https://mdigman.com/portfolio/twitterclassifier.html
[C]http://didtrumptweetit.com/machine-learning-nltk/
[D]https://cs.nyu.edu/grishman/jet/guide/PennPOS.html
[E]https://keras.io/examples/imdb_cnn/
[F]https://machinelearningmastery.com/develop-n-gram-multichannel-convolutional-neural-network-sentiment-analysis/
[G]https://www.aclweb.org/anthology/E17-2106.pdf

# Appendices

## Appendix 1: Real Trump Tweet Samples

- Vote @MikeGarcia2020 by May 12th! His opponent @ChristyforCA25 voted to fire teachers, and cut their salaries while bumping up her pay. Now she's mocking our Great Vets! We need Navy Fighter Pilot Mike Garcia in #CA25!
- I was thrilled to be back in the Great State of Arizona with the incredible, patriotic, and hardworking men and women of @Honeywell!
- OBAMAGATE!
- Hopefully our Country will soon mend. We are all missing our wonderful rallies, and many other things!
- Many complaints coming in about Maine. I love that State, won Maine 2 - Don't make the cure worse than the problem itself. That can happen, you know!

## Appendix 2: Real Teigen Tweet Samples

- Activity of the day was to make something super cute that you don't even like - because kids
- I think we are alike in so many ways. I remember the exact time I realized I wasn't allowed to say whatever popped in my head-that I couldn't just say things in the way that so many of my friends were saying. Before, I never really knew where I stood in the industry, in the world
- Lol! <3
- Officially saying farewell to baby miles and hello to toddler miles this week!!! oh how I've waited for this. finally calling me mommy and almost kind of speaking. And utensil usage ahhhh
- thank u!!!

## Appendix 3: Fake Trump Tweet Samples

- Where in the World is Joe Biden?
- Article IOTA has the power to overwhelm the corrupt and dishonest Federal Government. It IS our Money. We are building a REAL of the Next GENRE Economy in Europe. We are a Infrastructure Superpower that will NEVER give up our Texas, Ohio, Pennsylvania, New Jersey, Delaware, Conne
- Thank you Martha! #PrayForTheReds

- Thanks Jen, you are an incredible man and have my complete and total support. We will have a GREAT! MT @AlanDix 2 1/2 Weeks to go!
- I will tell you that Andrew is an incredible coach and motivator! He is one of our most influential voices in terms of bringing our youth directly into the adult world. He is an amazing guy to have around the D.C. area! @Alexandalexander @MPFaustion NEW MATTER PICKS!

## Appendix 4: Fake Teigen Tweet Samples

- I love you too!!
- Sometimes, ive been too drunk to read the comments. Please don't judge.
- General tidbit: we don't have a ton of staff, and there really isn't anything else to do except hope they are all ok. And yeah. it's so so very sad
- Do u want some eggnog???
- It's been awesome flying around the world with you guys, Patrick. I couldn't not hug you both @ThePatriciaReid "Luna"