Understanding Machine vs Human Generated Text in News

Black Lives Matter (US Category) `CSE 472 - Group 3`

By: John Oper & Joshua Dipert

Introduction

For our project, we observe the differences between articles written by humans and articles that were generated by a GPT-2 trained machine model. This consisted of scraping CNN's website for articles in the US related to the topic we were assigned, which for us was "Black Lives Matter". After the data was collected, we were able to begin generating machine text based on the articles that were previously scraped. Once we had machine generated text for each article, we were able begin observing the differences between the two. This notebook describes our processes, our methodologies our results, and more.

Literature Review

For the data scraping portion of our project we used the example script provided on github and we also read about the functions available in the beautifulSoup library for python3. Cited here, for the example script, and here for the beautifulSoup documentation. For the second step of the project we used the documentation provided for training and producing output from the GPT-2-simple model found here. This in addition to the Google Colab were used as a guideline for a model to train our machine in order to produce machine generated text based off of our initial data set.

Deliverables

The package `gpt-2-simple` is an open source text generating software that we used to train and generate our GPT-2 model

- This model was created by Max Woolf at MIT

- The package and documentation can be found here on Github

Getting Started

The code below downloads all the necessary packages used in this project

Notes:

- `gpt-2-simple` is only compatible with Tensorflow 1.x
- `Tensorflow` 1.x cannot be ran on Python 3.8.x

More on the Code & Data

To run the code we used for this project, you can do the following:

- Make a copy of this collab to run each block (note that this will take days and is not recommended)
- Add the necessary packages to your own machine and run the files attatched in the zip file for the project submission or found here in the github repository

More on the functionality, implementation, and reasoning of the code can be found throughout this document and in the code comments

The data used for this project can be found in the ZIP File with the project submission, in this github repository, or in the contents folder in Google Collab

```
# Deliverables
!pip install selenium
!pip install pandas
!apt-get update # to update ubuntu to correctly run apt install
!apt install chromium-chromedriver
!pip install -q git+https://github.com/huggingface/transformers.git
!pip uninstall tensorflow
!pip install tensorflow==1.13.2
!pip install -q gpt-2-simple
```

Methodology

Extraction Reasoning

Using Only CNN's Website

All of the data used for this project was extracted from CNN.com. We decided to extract from only CNN's website and not other's for the following reasons:

- CNN allows for an easy way to search for Black Lives Matter articles that are only in the US category, which was a requirement for our category
- CNN is a verified news source and has an ample amount of articles on Black Lives Matter
- Pulling from other sites would not have had little to no impact on the results, and could have potentially given worse results
- The code becomes unnecessarily complex when having to parse the html of different sites
- Allows for more uniform writing style when training the model, so the model sounds more cohesive when creating articles of its own
- Limits the amount of articles relating to one topic
    - Minimizes contradicting articles on the same topic
    - Maximizes the amount of fake news the model will create

The Extraction Process

Searching & Extracting the Article Hyperlinks

In order to extract the data from CNN, we had to use a `webdriver`. This is becasue CNN's site uses Javascript to display their site. Using a Chrome webdriver was the simplest solution we found. To search for the initial articles relating to Black Lives Matter in the US, we used the following search link "https://www.cnn.com/search?size=100&q=black%20lives%20matter&category=us&type=article&from= " where

- `size=` allows for n results to be pulled at once, we set n as 100
    - This is also the maximum size we found that CNN's site allows
- `q=` is the text that the site will query for, in out case it was Black Lives Matter

- `category=` is the category that the reult is queried from, this was set to us
- `type=` is the type of result queried for, we were only interested in articles so we set it to articles
- `from=` is the nth article to start from in the query pull
    - This was iterated by 100 in a for loop since we were pulling 100 articles at once
    - When we initially pulled the articles, there was a little over 800 articles, so the loop was ran nine times

The html results from each search were partially parsed and stored in an array. Once all results were stored, the array was iterated through where each link was fully parsed and stored in a different array to be used in the next step.

Extracting the Headline and Body of Each article

Each article was visited from it's associated hyperlink, which was extracted in the previous step. The headline & text of the article were then extracted from the websites html. Since the prefix for the GPT-2 model is the headline of the article and since we only want the model to write articles on Black Lives Matter, we had to ensure that each healine related to Black Lives Matter. We were able to accomplish this by:

1. Lowering the text of the headline and storing it in a temporary variable
2. Seeing if the headline contained "blm" or "black lives matter"
    - Articles that contained either "blm" or "black lives matter" were stored in one array for the headline, and one array for the article text
    - Articles where the headline did NOT contain "blm" or "black lives matter", the text was stored in a .txt file to train the model (more on this in the next section)
    - If the headline contained "BLM", then the headline was changed to "Black Lives Matter" instead
        - This is because we were unsure whether or not the model would understand "BLM"

The previous methodology resulted in 95 articles whos headlines would be what the model would write about. It also resulted in over 700 articles about black lives matter to train the model on.

Exporting the Data

Using the `pandas` package, we extracted the 95 articles into a .csv file. The articles index, headline, and text were each put into their own column and associated row for later use. The other 700+ articles were stored in a .txt file to train the model.

```
from bs4 import BeautifulSoup as soup
from urllib.request import urlopen as uReq
import time
from selenium import webdriver
import pandas as pd

# Create a chrome driver to search CNN's website and get articles related to the search
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--headless')
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')
driver = webdriver.Chrome('chromedriver',options=chrome_options)

# Dictionary for getting articles from CNN
# The URL gets 100 articles at a time, this was the max that one page on CNN would load
cnn_dict = {
    "url": "https://www.cnn.com/search?size=100&q=black%20lives%20matter&category=us&type=article&from=",
    "domain_url": "https://www.cnn.com",
    "class": {"class": "cnn-search__results-list"}
}

all_articles = [] # Stores all results from the CNN searches

# At the time of creating this there were 10 pages of search results from CNN's website relating to Black Lives
# Multiplying j by 100 will search for 100 new articles each time it is called
# Sleep must be called to slow the requests down in order to pull all the necessary data
for j in range(0, 10, 1):
  url_request = driver.get(cnn_dict["url"]+str(j*100))
  time.sleep(2)

  html = driver.page_source
  page_soup = soup(html, "html.parser")
  domain_url = cnn_dict["domain_url"]
  site_class = cnn_dict["class"]
  text_sections = page_soup.find("div", site_class).find_all("a")
  all_articles.append(text_sections)


# Article Urls stores all urls found in the search, in order to store them later
```

```
# The following for loop gets the results of the previous search
# i is incremented by 2 to avoid duplicate results since CNN's wesite has 2 links stored under href
# https: is added in order to create a valid link
article_urls = []
for j in all_articles:
  for i in range(0, len(j), 2):
          t = "https:" + j[i].get("href")
          article_urls.append(t)


used_article_urls = []  # Stores the articled URLS that have "BLM" or "Black Lives Matter" in the headline
headlines = []          # Stores the headlines of articles that have "BLM" or "Black Lives Matter"
article_text = []       # Stores the content of the articled that have "BLM" or "Black Lives Matter" in the hea
training_text = []      # Stores the text from the articles that DO NOT have "BLM" or "Black Lives Matter" in t

# The following function iterates through the URLS that were scraped in the search on CNN and returns it back t
def text_f_html(read_html, html_func, parse_section):
    page_soup = soup(read_html, "html.parser")
    for i in article_urls:
         text_sections = page_soup.find_all(html_func, {"class": parse_section})
         joined_texts = ""
         for j in text_sections:
            joined_texts = joined_texts + " " + j.text
         return joined_texts

# The following loops through the article URL's collected in search and pulls the html text from each
# Each headline from the url is searched for the text "Black Lives Matter" or "BLM"
  # If the headline contains "BLM":
    # All instances of "BLM" in the header are changed to "Black Lives Matter"
    # The headline is stored to be the prefix of the GPT2 trained model
    # The text is store for data observation and analysis
  # If the headline does not contain "BLM" or "Black Lives Matter":
    # The text of the article will be used to train the GPT2 software
for i in article_urls:
    uClient = uReq(i)
    read_html = uClient.read()

    headline = text_f_html(read_html,"h1","pg-headline")
    temp_headline = headline.lower()
```

```
    if "black lives matter" in temp_headline or "blm" in temp_headline:
      headline = headline.replace("BLM", "Black Lives Matter")
      headlines.append(headline)
      article_text.append(text_f_html(read_html, "div","zn-body__paragraph"))
      used_article_urls.append(i)
    else:
      training_text.append(text_f_html(read_html, "div","zn-body__paragraph"))

# The following creates a text file of articles whos headline did NOT contain "BLM" or "Black Lives Matter"
file = open("/content/BLM_CNN_GPT2_training.txt", "w")
for i in training_text:
  file.write(i)
  file.write("\n")
file.close()

# Dictionary to create .csv file with all articles who contain "BLM" or "Black Lives Matter"
retrievedArticles = {
    'Article_Urls': used_article_urls,
    'Headlines' : headlines,
    'Article_Text' : article_text
}

#Data frame to export the articles as a .csv
df = pd.DataFrame(retrievedArticles, columns=['Article_Urls','Headlines','Article_Text'])
df.index.name = 'Index'
df.to_csv(r'/content/BLM_CNN_articles.csv', index = True, header = True)

print(df)

```

Index	Article_Urls	...	Article_T
0	https://www.cnn.com/2020/09/27/us/online-prote...	...	High-profile killings of several Black people
1	https://www.cnn.com/2020/09/26/us/texas-teache...	...	Lillian White, an art teacher at Great Hearts
2	https://www.cnn.com/2020/09/22/us/black-lives-...	...	But that support has declined since early Jun
3	https://www.cnn.com/2020/09/14/us/iyw-children...	...	The attorney, who's an avid reader, went onli
4	https://www.cnn.com/2020/09/18/us/blm-protests...	...	Doug Swartz, Canal Fulton's police chief of e

Methodology Continued

Creating The Model

The Why

To re-iterate, the model was created from only CNN articles and on all articles whos healine did NOT contain "BLM" or "Black Lives Matter". Each of these articles content was stored in a .txt file called 'BLM_CNN_CPT2_training.txt'.This file ended up consisting of over 700 articles, where each article was seperated by a new line. Since the point of this project was to understanding machine vs human generated text in news, we needed to ensure that the article was fake, was created by the model and not just copied from previous articles, and was as human-like as we could get it.

The decision of only training the model on articles that it would not be writing about was made for two reasons. The first being that we wanted the news to be “Fake News”. In order to ensure that it was not fake news, we didn’t want it to be trained on the topics it was writing about. Of course there are articles written on the same topic with a different headline, but we wanted to minimize this as much as possible. Note that we were also to able to help minimize this by only pulling articles from CNN. The next reason for only training the model on articles that it would not be writing about is to maximize the likely-hood that the software create it's own original text. If the model were trained on the exact article, then it could just copy and paste the text from the actual article. Although thsi could still happen, we wanted take all the precautions we could think of to make sure that it didn't. This was also validated on a small scale by taking little pieces of text from the models output and searching for it on the articles that it was trained on.

At first we had ran some of the article headlines on the pre-trained '755M' model from gpt2-simple. While these results were good, they didn't seem human-like and also tended to veer off topic very quickly. By the end of the article, the text generator had written about a completely different topic in many cases. To get a better output, we decided to train the model ourselves. This resulted in much better reults. Also, training the model on only CNN articles allowed for the model to be more fluid throughout since the formtting was similar and since the number of journalists the model was trained on was minimized.

The How

The model was trained by first importing the '355M' model from gpt-2-simple. Accorinding to their documentation, this is the "medium" model. There is one model smaller and two models larger. We did not have a machine that could train any larger models, which is why we chose this model. This model was trained from the .txt filed called `BLM_CNN_GPT2_Training_V1.txt`. It took around 8 hours to train and resulted in a 1.6GB model. The following parameters were set to finetune the GPT2 model:

- `steps = 1000` seemed like a good number of steps to train the model on, we could have done more or less, but this seemed like a good middle ground
- `restor_from = fresh` we wanted to train a model only from our articles so we wanted to have a fresh model
- `run_name=run1` is the folder that the run is stored in
- `print_every=1` it made it seem faster when it printed more often
- `sample_every=100` we were able to see an example every hundred which helped validate that our model was improving
- `save_every=100` we had it crash at 380 steps once and we had it saving at 500 steps so we lowered that to have a some-what close save

`WARNING:` *Running the below code can crash your computer. Please ensure that your computer is capable of running this code. The code will also take awhile to run unless you have a powerful computer so it is reccommended to just use our trained model instead of training a new one`*

```
import pandas as pd
import gpt_2_simple as gpt2
import os
import requests

# 355M is the medium model and is the largest model that could be ran in google collab and our personal machine
model_name = "355M"
if not os.path.isdir(os.path.join("models", model_name)):
    print(f"Downloading {model_name} model...")
    gpt2.download_gpt2(model_name=model_name)   # model is saved into current directory under /models/355M/

# The file created from articles that did NOT contain "BLM" or "Black Lives Matter" in the headline
file_name = "BLM_CNN_GPT2_Training_V1.txt"

```

```
# Train the model from the text file in the previous step
sess = gpt2.start_tf_sess()
gpt2.finetune(sess,
              dataset=file_name,
              model_name="355M",
              steps=1000,
              restore_from='fresh',
              run_name='run1',
              print_every=1,
              sample_every=100,
              save_every=100
              )

# generate the model created
gpt2.generate(sess)

```

Methodology Continued

Running the Model

The model was ran by first importing the .csv file called `BLM_CNN_articles.csv` by using the pandas package. This is the .csv file that was created in the first step. Each column of this data was then stored in an array to be later uploaded into a final .csv file. Generating the model was first done by looping throgh the headlines from the array created. Then, the words of each article text associated with the headline was counted and then used to set the tokens that the model would write about equal to the number of words in the initial article. This was to show similar results from each article. The `temperature` and then `top_p` were both set to the values recommended in the documentation. The results from each genterated article were ouputed in a list, and then each of these values were appended to an array. Each generated article was also written to a text file called `BLM_CNN_GPT2_ouputs.txt`. The index, article urls, headlines, article text, token length, and generated text were exported into a .csv file called `BLM_CNN_GPT2_Output_articles.csv`. This allowed for an easier way to compare and observe the original verses generated text.

```
import pandas as pd
import gpt_2_simple as gpt2
```

```
import tensorflow as tf
import os
import requests

# Reads the .csv file created from searching for articles on CNN's website
df = pd.read_csv('BLM_CNN_articles.csv')

# Store the data extracted from the .csv file
article_urls = list(df['Article_Urls'])
headlines = list(df['Headlines'])
article_text = list(df['Article_Text'])

# Creates a new sesstion to run the model trained in the previous step
sess = gpt2.start_tf_sess()
gpt2.load_gpt2(sess, model_name='run1')

gpt2GeneratedText = []
gpt2_length = []
file = open("BLM_CNN_GPT2_ouputs.txt", "w") # Stores the output of each GPT2 generated article
for i in range(0, len(headlines), 1):
    print("----------- " + str(i) + " -----------" )
    headline_prefix = headlines[i]

    # Set the length of the generation to the max generation length if the article length is greater than the m
    articleLength = len(article_text[i].split())
    if articleLength > 1023:
        articleLength = 1023
    output = gpt2.generate(
                sess,
                prefix=headline_prefix,
                length=articleLength,
                temperature=0.7,
                top_p=0.9,
                return_as_list=True
        )
    print(output)
    gpt2GeneratedText.append(output[0])
    gpt2_length.append(articleLength)

```

```
    file.write("---------------------------- " + str(i) + " ----------------------------\n")
    file.write("Prefix Text (Headlines): " + headline_prefix + "\n")
    file.write("length: " + str(articleLength) + "\n")
    file.write("GPT2 Output:\n" + output[0] + "\n\n")

file.close()

# Dictionary to storte the collected data in a .csv file
retrievedData = {
    'Article_Urls': article_urls,
    'Headlines' : headlines,
    'Article_Text' : article_text,
    'Generated_Text': gpt2GeneratedText,
    'Generated_Text_length': gpt2_length
}

#Data frame to export the articles as a .csv
df = pd.DataFrame(retrievedData, columns=['Article_Urls','Headlines','Article_Text','Generated_Text','Generated
df.index.name = 'Index'
df.to_csv(r'BLM_CNN_GPT2_Output_articles.csv', index = True, header = True)

print(df)

```

Data Set

The data set that the model wrote about consisted of 95 articles from CNN relating to Black Lives Matter in the US section. The data set that the model was trained on consisted of over 700 articles from CNN relating to Black Lives Matter in the US section. The final set of generated text was stored in a .csv and a .txt file. A more detailed breakdown of the name, description, and what the file consists of can be found below. A more detailed description of the data set can be found in the `"Methodology"` section of the report.

Initial Data Sets

- Name: `BLM_CNN_articles.csv`
- Description: This data set is the result of all articles whos headlines contained "blm" or "black lives matter"

- Usage: Allowed for the GPT-2 model to generate text based on these values
- Contents:
  - `Index` the array value that the data was stored in
  - `Article_Urls` the url for each article
  - `Headlines` the headline of each article
  - `Article_Text` the text of each article

- Name: `BLM_CNN_GPT2_Training_V1.txt`
- Description: This text file consits of all the articles whos healines did NOT contain "blm" or "black lives matter"
- Usage: Trained the GPT-2 model
- Contents:
  - 700+ articles text with a new line between each article

Model

- Name: `checkpoint -> run1` (folder)
- Description: The model generated from the text input
- Usage: Generated text based on the headlines

- Name: `samples -> run1` (folder)
- Description: Sample outputs for every 100 steps while training

Final Outputs

- Name: `BLM_CNN_GPT2_Output_articles.csv`
- Description: This data set is the result of `BLM_CNN_articles.csv` combined with the associated GPT-2 generated text and the number of tokens used for the generated text
- Usage: Observe and compare results
- Contents:

- `Index` the array value that the data was stored in
- `Article_Urls` the url for each article
- `Headlines` the headline of each article
- `Article_Text` the text of each article
- `Generated_Text` the text generated from the GPT-2 trained model
- `Generated_Text_Length` the number of tokens used to generate the GPT-2 text

- Name: `BLM_CNN_GPT2_ouputs.txt`
- Description: Text file with all the outputs of the GPT-2 generated text
- Usage: Observe the text by itself and search for any copied text (none found)
- Contents:
  - `index` the index that it is associated with in the `BLM_CNN_GPT2_Output_articles.csv` file
  - `Prefix Text` the headline of the article / prefix text of the generated text output
  - `GPT-2 Output` the generated text output associated with the text file

Observations & Discussion

The GPT-2 generated text gave some very impressive results and we would go as far to say that a large amount of the articles could trick a Human Being. The overall style, formatting, punctuation, and spelling is constent with that of the initial article. One of the more impressive things is how the machine generated text was able to create realistic sounding quotes along with names, and stastics that are of course completely fake. An example this can be found on index 5 of the .csv or .txt file. It states:

Public support for the Black Lives Matter movement has dropped since June, report finds. Meanwhile, the number of Black Americans who say the police are doing a good job has increased from 35% in June 2016 to 49% in a new CNN/ORC poll. It's a change that's been driven in part by increased support for Black Lives Matter. "It's not surprising that the black community is seeing more positive messages from the media and politicians about police officers," said Randal O'Toole, a professor of African-American studies at the University of Missouri-St. Louis. "The public's response to the unrest in Ferguson has been to push for more police involvement in the community and to demand justice for those killed by police." The city's police chief said

the decision to fire Wilson was "not the right decision" and that he would "look at all options to resolve this matter." "Our officers have done what they have to protect themselves and their communities, and I believe the actions of our police officers were appropriate and consistent with our commitment to the safety of our citizens," Police Chief Gary W. Wilson said in a statement. "I am confident that we can all agree that there are things we all can agree on -- that our community can agree on -- and that is that there must be a full investigation of this incident." The Ferguson Police Department has been under fire since the fatal shooting of 18-year-old Michael Brown in August of 2014, and the resulting protests and national attention have brought racial tensions to the fore. Protesters have called for justice in the killing of Brown and for the formation of a police oversight board to oversee the department. "The Black Lives Matter movement has had a great impact on the conversation and change in Ferguson," said James E. O'Neill Jr., a political science professor at the University of Missouri-St. Louis. "The Black Lives Matter protests have brought about a lot of change in how we see the police department, but what's missing is the critical component that Ferguson needs -- the police officer."

This is not just a one-off article either. You are able to find plenty of text generated articles that are this good, if not better. The overall the content of the machine generated text is fluid and makes a lof of sense throughout. It doesn't seem to veer to far off topic and from start to finish is consistent. If we were someone who was completely unaware of anything related to the Black Lives Matter movement, we might actually believe some of these articles, which is kind of scary to think about. Being aware of the actual topic definitely helps point out the machine generated from the real articles. One machine generated article that stood out to us was index 31 that said:

Atlanta's WNBA team supports Black Lives Matter after pushback from co-owner, a US senator, and several other NBA players. NBA Commissioner Adam Silver said he has "zero tolerance" for racism in the league and he issued a statement calling on NBA players to "empowerment and uplift all of our communities." "The NBA stands behind our players and employees who play the sport we love," Silver said. "We stand with our players and employees who are part of our communities. That is the spirit that made us the world's greatest." "The NBA is an inclusive place, and we welcome all people of all backgrounds, races, nationalities, and religions to play and be part of our league," Silver said. "We support our players' right to play in whatever way they want."

We thought that some of these were real quotes and it seemed like a copy and paste since we were unaware of everything NBA or WNBA have said about the BLM movement. The quote "We stand with our players and employees who are part of our communities.

That is the spirit that made us the world's greatest." could not be found in the articles used to train the model, or any of the other articles. We searched bits and pieces of the text and could only find a few pieces of text that were close but did not have the same meaning. The overall content of the machine generated text sounds like it comes from a trustworthy news source, and if you didn't understand or know anything about the topic, you would probably believe some of these articles.

The main difference between human and machine generated text is that the machine generated text tends to repeat certain words or phrase more often then that of the human. It also tends to start a lot of sentences with the same word and some of it's quotes are inconsistent. For exxample, many quotes and instances of information that are cited have a generalized source or reference, such as a "a neighbor", or "a woman", but even something like that could be due to a request of anonymity from the participant or whoever was questioned. We concluded that based on examining the news articles, we would assume that a fair amount of these articles would be written by a human if we didn't know which was which.

The thing about this is that this was only generated by the "medium" model, with only 1000 steps, only 700+ articles and with only out sample of each generated article. It poses the question of how much better would this model be if it were trained on a bigger model or trained for longer or had more articles about black lives matter or had more samples to choose from? How good can this actually get? We believe that if someone figured out the correct formula for this then there this machine generated fake news would be undetectable without an AI or without being educated on the topic.