CSE 472: Social Media Mining

Understanding Machine vs. Human generated text in news

Phase 1 Project Report [Group - 2]

TOPIC: Election 2020

GROUP MEMBERS:

First Name	Last Name	ASU ID
Neha	Tandon	1216798336
Prashasthi	Mavinakunte	1217002098

CONTENTS

1. INTRODUCTION

Machine generated text or artificial text is in abundance on the internet right now. This project focuses on web crawling to extract information on election 2020 data from news websites and performing exploratory analysis on the extracted text and the machine-generated text.

1.1 Background:

Text generation has been an ongoing research for ages now, it includes open domain text generation, domain text generation, style specific generation, news generation, etc. The models we have used in the project are from the GPT2 Huggingface library which is for open domain text generation. The text generation comes under natural language processing (NLP). NLP has three main steps: text pre-processing, text representation and analysis & modeling.

In the phase I of the project, we are creating a dataset with news based on 2020 elections consisting of headlines, human generated articles, machine generated articles, news source and model for text generation. The machine generated articles were constructed by using the headline as the input and we analyzed the distinction between the human generated text and machine generated one. The model used to achieve this is the GPT-2 model(Generative Pretrained Transformer 2).

1.2 Motivation:

What made the full version of GPT-2 particularly very strong was the way it could be “fine-tuned.” Fine-tuning involves a second round of training on top of the general language skills the machine has already learned from the Reddit data set. Feed the machine Amazon or Yelp comments, for example, and GPT-2 could spit out phony customer reviews that would skew the market much more effectively than the relatively primitive bots that generate fake reviews now, and do so much more cheaply than human scamsters. Pump-and-dump stock schemers could create an A.I. stock-picker that writes false analyst reports, thus triggering automated quants to sell and causing flash crashes in the market. Fake news would drown out real news.

The motivation of understanding the machine generated text and human generated in the news domain is to figure out if we could use GPT-2 to write a *New Yorker* article. That was our solipsistic response on hearing of the artificial author’s doomsday potential. OpenAI fine-tuned GPT-2 on *The New Yorker’s* digital archive—millions of polished and fact-checked words, many written by masters of the literary art could be faked. Could the machine learn to write well enough for *The New Yorker*? Which is why we have chosen to use the new york times news for our dataset so we could find out the answer to these questions at the end of our project.

2. LITERATURE REVIEW

Our literature review consists of mainly two references, the first one is for web scraping using Python libraries and the second one is of the model for machine-generated news articles.

In [1], we referred to the algorithm and code for web scraping to extract the news articles from the New York Times website for our dataset. The BeautifulSoup Library from Python is used to extract a list of news articles based on the search query for a given topic. The HTML structure of the webpage is used to find the element which contains the link of the article. This may vary for all websites. The links are then extracted for all the articles and saved in a list. The Client object then parses through each link to extract the headline and the body of the articles. This also uses the HTML structure of the webpage that contains the article to extract the information we want to scrape and the results for each article are stored in a csv file.

In [2], we learn about the existing API: HuggingFace Pytorch Library for text generation.
This article briefly breaks down the decoding strategies and their implementation using the transformers library. This library provides functions for auto-regressive language generation, that is based on the assumption that the probability distribution of a word sequence can be represented as a product of the conditional distributions of the next word(Fig 1).

$$P(w_{1:T}|W_0) = \prod_{t=1}^{T} P(w_t|w_{1:t-1}, W_0) \text{ ,with } w_{1:0} = \emptyset,$$

Fig 1: Probability distribution of a word sequence[2]

Where W_0 is the initial context word sequence and T is the length of the word sequence, which is tuned according to requirements and corresponds to the timestep t = t and the EOS token is determined from:

$$P(w_t|w_{1:t-1}, W_0) \text{ [2]}$$

The author explains the decoding methods: Greedy search, Beam search, Top-K sampling and Top-p sampling.

The Greedy search algorithm chooses the word with the highest probability in the sequence as its next word at each timestep. The disadvantage of this method is that it misses high probability words hidden behind a low probability word[2]. Beam search takes care of this problem by tracking the most likely beams sequences at each timestep and chooses the sequence with the overall highest probability at time t. However, there might still be some repetition, as is in greedy search. To solve this problem, a parameter is introduced that ensures that any word sequence of n words is not repeated more than a certain number of times[2].

The author also discusses sampling methods; namely: Top K-sampling and Top p-sampling. In Top-K sampling, the K most likely next words are filtered and the probability mass is redistributed among only those *K* next words[2]. However, T*op-K* sampling is not dynamic and does not adapt to the number of words that are filtered from the next word probability distribution. Thus, the results of this method depends on the sharpness of the probability distribution, that is, if some words are taken from a sharp distribution, the method produces incoherent gibberish, as compared to a relatively normal distribution[2]. This brought us to the Top-p sampling method, which chooses the smallest possible set of words whose cumulative probability exceeds the probability *p*. The probability mass is then redistributed among this set of words. This way, the number of words in the set dynamically increase/decrease according to the next word's probability distribution[2].

The review of these decoding methods led us to examining the results of these methods in our context of news article (text) generation.

3. METHODOLOGY

We created our dataset in two steps. In the first step, we extracted news articles for the topic of Election 2020 from the New York Times website (https://www.nytimes.com/). To create our dataset, we scraped the headline, the article and the source (article link). For the second step, we generated the machine article by giving the same headline as input and running the HuggingFace library model using the beam search algorithm to generate the body of the article.

headline	human-generated text from news article	news article source	machine-generated text for same headline	source/model for machine-generated text

Table 1: The dataset format

Table 1 shows the structure of the dataset that we generated. In step one, we wrote a web scraping script in Python to automate the human-generated news extraction process[1].

For our chosen website (https://www.nytimes.com/), the search results show 10 news articles per page, along with a load more button that displays the next 10 articles every time it is interacted with. However, the URL remains the same. For the purpose of extracting all the 80 articles by running our script only one time, we used the Selenium Webdriver existing API in Python, which is a tool that allows browser automation for testing and other purposes[3]. This allowed us to interact with the load more button in an automated manner through our code, so that we could extract the articles after at least a 100 of them are loaded (10 clicks). By using the XPath of the last required article in our list, we controlled the wait time before the scraping of the elements[4].

After the required number of articles were loaded in the list, we used BeautifulSoup library in Python[1]. The HTML structure of the webpage was used to find the element which contained the links of the articles. The links were then extracted for all the articles and saved in a list. The Client object then parsed through each of these links

to extract the headline and the body of the articles. We used the HTML structure of the webpage that contains the article to extract the information we want to scrape here as well and the results for each article were stored in a csv file[1].

In step 2, [2] in order to the load the model required, installation of transformers was done. GPT2 in Tensorflow 2.1 is the module being used for text generation. The headlines extracted from the news website into the csv were fed as an input to the GPT-2 module. Initial trial was performed with greedy search which clearly generates a text that starts repeating itself. The reason was because it was missing the high probability words hidden behind a low probability word. After this observation, the choice of beam search was made. In beam search, the risk of missing hidden high probability word sequences is reduced by keeping the most likely num_beams of hypotheses at each time step and eventually choosing the hypothesis that has the overall highest probability. While the result is more fluent, the text still includes repetitions which was minimized to an extent with a simple remedy called n-grams (a.k.a word sequences of n words). It makes sure that no n-gram appears twice by manually setting the probability of next words that could create an already seen n-gram to 0 . The **no_repeat_ngram_size=2** and **early_stopping=True** so that generation is finished when all beam hypotheses reach the EOS token. In addition to the above, a trial with *Top-K* sampling where the *K* most likely next words are filtered and the probability mass is redistributed among only those *K* next words.However, an observation was made that the sampling scheme suffered from generating repetitive word sequences and incoherence. This is because of the way the model is trained[2]. Hence, the decision of using beam search with ngram size as 2. The csv we obtained was fed with the machine generated texts for the corresponding headlines.

4. DATASET

The dataset is a table of 5 columns and 81 rows. The columns are: headline, human-generated news article, source for human-generated news article, machine-generated news article, model/source for machine-generated news article. As mentioned above, we have taken our human generated news from the nytimes website. Here is an example of an entry(row) in the dataset: Table 1 gives columns 1 and 2 of the dataset, Table 2 gives columns 4 and 5 of the dataset. Column 3 of the dataset contains the source of the human-generated news article, like [5].

Headline	Human-generated news article
Democratic Donors Push Biden for a Cabinet Free of Fossil Fuel Connections	WASHINGTON — Liberal activists have opened a new front in their battle to push Joseph R. Biden Jr. to the left on climate change: a campaign to pressure the Democratic presidential nominee to reject advisers with any ties to fossil fuel companies.The targets include former Energy Secretary Ernest Moniz, former Interior Secretary Ken Salazar and others who served in the Obama administration.Last week more than 60 deep-pocketed donors asked Mr. Biden to commit to a moratorium on all new coal, oil and natural gas development — and to select advisers who are “free from fossil fuel influence.” The group, which includes Kathy Washienko, a clean-energy investor in Seattle;

	Robin Chase, a co-founder and former chief executive of Zipcar; and Adelaide Park Gomer, the president of the Park Foundation philanthropic group and a longtime anti-fracking activist, implored Mr. Biden to “choose new, bolder leadership” than those with whom he worked under President Barack Obama.“This incumbency of old ideas (like ‘all of the above’ energy policy) must end,” the donors wrote.The effort comes just days after Mr. Biden denounced President Trump as a “climate arsonist” and made the case that Americans suffering from wildfires, floods and hurricanes could not afford four more years of a climate-change denier in the White House. Even many of Mr. Biden’s persistent critics on the left described it as the most aggressive and detailed policy speech on climate change ever given by a presidential candidate.

Table 2: Example of Human-generated news headline and article taken from [5]

Machine-generated news article	Source
Democratic Donors Push Biden for a Cabinet Free of Fossil Fuel Connections Biden, who is running for the Democratic nomination for president in 2016, has been a vocal critic of the fossil fuel industry. In an interview with The Huffington Post, he said, "I don't think it's fair to say that we're going to have a clean energy future. I think we need to do something about it." He also said that if he were elected president he would "take a hard look at what's going on in this country and what we can do to make sure that it doesn't happen again."	Huggingface

Table 3: Example of Machine-generated news article for the same headline (taken from Table 2), generated through the Huggingface Library in Python

5. OBSERVATIONS AND DISCUSSIONS

There is a considerable difference between human and machine generated news articles in our dataset. The human-generated articles make much sense as compared to the machine generated ones, that often produce incomprehensible gibberish. The content does not match with the headline and it generated some random links in the body of the article. One can easily recognize which article is human-generated and which is machine-generated simply by going through the content. The machine generated articles lack logical sense and in some cases even semantics are not proper. For the following parameters, here are our observations:

- **Style -** Is the style of the generated article consistent?

The style of the machine generated article is not consistent, it contains random information for certain articles and links to images or tweets that do not exist. One of the examples picked from the dataset that shows this is:

Trump Health Aide Pushes Bizarre Conspiracies and Warns of Armed Revolt

In an interview with ABC News' George Stephanopoulos on "Good Morning America," Sen. Bernie Sanders (I-Vt.) said, "I don't think there's any question that we're going to have an armed revolution in this country. And I think that's what we've got to do. We have to make sure that the American people know that this is not just a political issue. This is an economic issue."
.@SenSanders: "We have a lot of people who are not prepared to stand up for what they believe in." pic.twitter.com/QJ6YqY3qXJ — ABC Politics (@ABCPolitics) November 8, 2016
(RELATED: Sanders: 'We Have a Lot of People Who Are Not Prepared To Stand Up For What They Believe In')

- **Content -** Does the content of this article make sense?

  The content of the article is incoherent with the headline and is producing gibberish in some cases.One of the examples picked from the dataset that shows this is:

Can Neil Young Block Trump From Using His Songs? It's Complicated

In a recent interview with the New York Times, Young said, "I don't know if I'm going to be able to do it, but I do know that I have a lot of respect for Donald Trump. I think he's a great person, and I respect him for that. But I also think it's important for him to get out there and say, 'Hey, you know what? I want you to listen to my music.' And that's what I've been doing for a long, long time. That's why I love him so much. He's an amazing person. And I hope that he can do the same for me."

- **Overall-** Does the article read like it comes from a trustworthy source?

  No, the articles do not look like they are coming from a trustworthy source. The mismatch between the headline and the content of the article when compared with the original (human-generated) article is significant for many cases in our Dataset.One of the examples picked from the dataset that shows this is:

Trump Defies Nevada Directive as Thousands Gather for Indoor Rally

The Nevada Supreme Court ruled on Tuesday that the state's ban on same-sex marriage violates the U.S. Constitution's guarantee of equal protection under the law. The ruling is the latest in a series of rulings by the high court in recent months that have raised questions about the constitutionality of a state law that bans gay and lesbian couples from marrying in public places.

As mentioned in [2], high quality human language does not follow a distribution of high probability next words. That becomes a clear identifier in this case, to separate human generated text from machine generated text. As humans, we want generated text to surprise us and not to be boring/predictable[2], which the machine-generated articles clearly are here.

6. DELIVERABLES

There are 2 code files and a csv dataset file in our submission.

1. *Smm_main_step1.py:* The first code file is a python script that extracts human generated news articles from https://www.nytimes.com/ for the search topic election 2020. It iteratively loads the website (by clicking the load more button 10 times), to show a total of around 100 articles and then scrapes the links for these articles. The links are then stored in a list, this list is then used to extract the headline and body of each article and the results are input into a csv (comma separated values) file.

2. *Smm_main_step2.py:* The second code file is a python script that gives the machine generated article for each article extracted in File 1, *Smm_main_step1.py* (that was stored in csv and is read by this file), by taking the headline of every article as its input. We use the HuggingFace Python Library and beam search decoding method to generate the machine article for every headline and the results are then stored in a csv file.

3. *Smm_dataset.csv:* We combined the results from both these scripts to create our dataset in the format as shown by Table 1. The dataset was cleaned manually to remove articles that were extracting incomprehensible language/symbols or not generating machine articles and our final dataset consists of 81 row entries and 5 columns, stored in a csv file Smm_dataset.csv (format as in Table 1). The delimiter is comma.

7. REFERENCES

[1] https://github.com/AmritaBh/CSE472_Fall20_Files/blob/master/scrape_news.py
[2] https://huggingface.co/blog/how-to-generate
[3] https://testguild.com/selenium-webdriver/#:~:text=Selenium%20WebDriver%20is%20an%20open,where%20you%20need%20browser%20automation.
[4] https://stackoverflow.com/questions/52800174/clicking-more-button-via-selenium/52804725#52804725
[5] https://nytimes.com/2020/09/22/climate/biden-climate-change.html?searchResultPosition=10