Theodore Hui

Asif Mahdin

Luke Taylor

Natalie Young

Classification of AI-Generated News Article Descriptions

Introduction

The proliferation of low-quality news articles in recent years has been a cause for concern for many due to the rise of social media and increasing reliance on internet-based news. With increasingly sophisticated large language models (LLMs), AI-written descriptions of these articles may be generated in such a way that they are more often found among results from search engines and article recommender engines than they would have been otherwise. Although there are many positive applications of LLMs, those who produce malicious content may optimize their content to be selected by such algorithms rather than prioritizing the quality of the content. We propose our project to build a classifier that can identify news article descriptions written by GPT-3 or other LLMs versus human-written text. While this in itself is not a novel idea, our method of generating data for the classifier may be considered as such.

Model

Originally designed by Google and deployed by the AI company Hugging Face in 2018, Bidirectional Encoder Representations from Transformers (BERT) is a neural network language model. Its architecture is transformer-based and is a deep learning model designed to process and understand sequential data, such as natural language text. In the BERT model, each word in a

sentence is represented by a high-dimensional vector called a *word embedding*. These word embeddings are then fed into the model, which processes the sequence of words in a bidirectional manner, meaning that it takes into account the context of words both before and after the word currently being processed in the sentence. This allows the model to capture the meaning and context of words in a way that is similar to how humans understand language. After word processing, the BERT model uses a multi-layer transformer architecture to learn the relationships between the words in a sentence and generate a sentence representation that captures the overall meaning of the sentence. This sentence representation is then used to perform a variety of natural language processing tasks, such as language translation and text classification. (ChatGPT)

Our model is an adaptation of BERT that is specifically tuned for text classification and appropriately named "BertForSequenceClassification." Essentially, it is the standard BERT model with a linear classification layer attached at the end. The full description of the layers in the model is linked here and referenced below (also available on the notebook). Instead of tuning a complete model from scratch, we opted to use one of Hugging Face's pretrained models ('bert-base-uncased') and fine-tuned it for our task. *As an aside, the task of fine tuning would have taken roughly 20 hours to complete using on our local CPUs, but only took a mere eight minutes when run on one of DataHub's 1080 Tis.*

Datasets

The dataset we trained our model on consists of roughly 11 thousand English news articles which were published on the internet by large, international written media sources. They were collected by Szymon Janowski between 2019-09-03 and 2019-11-04 and published on

Kaggle for public use. Although it should be noted that the dataset was collected with the purpose of analyzing reader engagement, the various attributes included for each article (which could be used as features for machine learning tasks) allowed for it to be used for our purposes: we only needed the URL link to the original news article and the article description.

Having collected the positive set—descriptions (presumably) written by humans or news editors—, we needed to generate the negative set of our dataset: descriptions produced by a language model. To achieve this, we split the dataset into chunks amongst group members and then wrote a script for processing them. The script sends a request to GPT-3's 'text-davinci-002' engine from our respective locally-run notebooks to write a two-sentence description of the article located at the specified URL. Each of the GPT-3-generated descriptions is set to begin with the same two words as in the dataset description to ensure a certain degree of syntactic similarity. One thing of note is that we set the temperature to zero when making our API calls to GPT-3 (more on that later).

Each group member used their personal OpenAI API keys to call on GPT-3 in order to save on credits. The original dataset from Kaggle contained the contents of the entire article in its own attribute column; however, feeding the entire article to GPT-3 through an API request was too inefficient on such a large dataset and was very costly due to the amount of tokens being sent over to GPT-3 from each article. The process took several hours to complete for each group member. When it was completed, we concatenated four .csv files into one dataset (fulldata.csv).

Results

At first glance, GPT-3's generated descriptions were convincingly similar to the original descriptions. To have a more quantitative idea of its performance, we ran our data through various existing classifier models as a reference on how well our BERT masked language model performed relative to more standard machine learning models, such as bag of words and TF-IDF vectorizers with ridge and random forest classifiers. Each combination of these classifiers and vectorizers had similar performance with an accuracy rate of around 80 percent, thus setting the baseline for our BERT model. This accuracy was already much greater than we had originally expected, leading us to believe that there must be clear fundamental differences between the default news article description and the GPT-3-generated descriptions. We were concerned that BERT might not show much of an improvement on this task compared to our baseline models. However, BERT did perform better–12 percent better that is–than our baseline models with an accuracy of approximately 92% on the test data, showing that the model performed exceptionally on learning the differences between GPT-3 descriptions and the default descriptions.

We believe that all the models performed well on distinguishing the differences between the GPT-3-generated and default descriptions because GPT-3 probably used specific words more often than the article descriptions did, which led to a high confidence of which class the description belonged to. When we performed some data analysis on the two classes in the dataset, we noticed that the most common words distinct from the default descriptions in the GPT-3 dataset were words that did not convey meaning, whereas the more common words in the default description dataset distinct from the GPT-3-favored words were more promotional words of that news outlet (showing that some of the data had extra information promoting that news

outlet along the description). For example, in the GPT-3 data, the top words were ‘is’, ‘has’, ‘article’, ‘discusses’, and ‘the’, whereas the default description appears to use less of these words that don’t contain meaning. The most common words in the descriptions section were words such as ‘at’, ‘top’, ‘coverage’, ‘exclusive’, so it seems like the default descriptions contains more promotional content of that outlet (i.e. ‘subscribe to ABC News for exclusive content’) and the verbiage seems to exclude meaningless words that GPT-3 may use as part of formal grammar. For example, GPT-3 may say “the article discusses the speech of President Trump” whereas a default description may say “President Trump spoke at a summit in…”, showing that requiring the same two start words in the GPT-3 description as the two start words in the real description was not quite enough prompting to get GPT-3 to appear similar enough to the real description’s syntax to “fool” the model.

Low temperature in our GPT-3 API call may have been another factor in making the two groups easy to distinguish. Lowering the temperature causes GPT-3 to have less variation in its word choice, picking the words that the model is most confident will come next, instead of favoring variation. Perhaps this led GPT-3 to become more predictable in its syntax so the model could more easily pick up on the words it used. Thus, to further improve our project we could figure out how to prompt GPT-3 better to “challenge” our models more, to find the limits on how good large language models can mimic human language in this context, because, at this point, it doesn’t seem very close.

Conclusion

Detecting AI-generated news descriptions is becoming an increasingly important task as most people rely on the internet to receive the news and are not accustomed to fact-checking. To

prevent misinformation and overexposure to these sorts of news stories, a system that is able to classify these news correctly with a high accuracy and warn users would benefit many people. As we can see from our results, GPT-3 is proficient at generating news article descriptions but it is far from passing as human-written. Our model is already able to achieve this with a very high accuracy but for a working version of this model that can be generalized to all types of news, the model needs to be trained on an even larger dataset. Our model is best suited to be used with search engines or social media as they account for the greatest portion of news being circulated. Whenever a news article is posted or shared, it would be run through our classifier, which would detect whether the news article description is written by humans. In the event that the article description is classified as being generated by another language model, a warning could be issued to the user that the article description might be fake. Having such a system in place would help curb the spread of low-quality news articles to a large extent.

References

Hui, T. (2022). BERT Layers. Retrieved from https://docs.google.com/document/d/19PJCM0XFJtUk_MSP5aPXJ2PLTWZM2_tgrbWhwm6R8go/edit?usp=sharing

Hui, T., Mahdin, A., Taylor, L., Young, N. (2022). Linguistics 167 Final Project Code. Retrieved from https://drive.google.com/drive/folders/1APafxcrfkSi1ecCIjUB6K6jOZgwQIQNz

Janowski, S. (2019). Internet news data with readers engagement. Retrieved from https://www.kaggle.com/datasets/szymonjanowski/internet-articles-data-with-users-engagement?resource=download

Peirsman, Y. (2019). Text classification with BERT in PyTorch. Retrieved from https://github.com/nlptown/nlp-notebooks/blob/5c7b2b0fe088884bc21c3a3ee18b3c248f847dac/Text%20classification%20with%20BERT%20in%20PyTorch.ipynb