The Surprising Humanity of AI Generated Writing, Using AI Dungeon

Ryan Bautista
Hunter College
rbautista9962@gmail.com

Abstract

Generated Writing using the GPT-2 and the GPT-3 language models are becoming increasingly ubiquitous in usage, and differentiating what was written by a person and what was written by a program has become a nearly impossible task. Through previous research, programs are able to create swathes of text that, at a glance, are impossible to discern as AI generated. However, there is very limited research on programs that can do the one thing people cannot do: tell the difference between one and the other. Using stories co-generated from humans and AI Dungeon 2's Griffin model, which utilizes the GPT-2 language model, I have attempted to create a program that is capable of determining if any given sentence is human or AI generated. However, the results are ultimately inconclusive, as the program has failed to differentiate these two creators. My findings with this project has allowed me to determine that, given a story of this caliber, it is far more difficult to determine human and AI generated writing, than what is capable with the Natural Learning Toolkit.

Introduction

Writing has always been considered to be a human task, and for the longest time, that has seemed to have always been the case. There always have been different methods and technologies to make writing easier, from speech-to-text to translators. These things are undoubtedly useful, but there's always one thing that these things have lacked: input not from a human. A human speaks, and that person's words are put into text. A human types up a phrase in one language, and the program outputs it in another. Even a program like a translator must rely on dictionaries, all human defined, to return a translation. For language, only until recently have there been programs that can "think" on its own, and output a response that is not entirely reliant on the input put in.

The language model Generative Pre-Trained Transformer, or GPT, is one such program. As an autoregressive language model, it is primarily focused on combining the previous sentences it creates, as well as a bit of randomness and influences by the user, to create new sentences in the form of whatever language that needs to be generated, following the necessary grammar and sentence structure to be understandable, usually. Created by OpenAI, a company that specializes in Artificial Intelligence, its main purpose is to create writing. Of course, "writing" is a very broad topic, but for later iterations of the GPT, named GPT-2 and GPT-3, the Generative Pre-Trained Transformer started seeing commercial and even recreational use. One such use is writing articles for newspapers and news sites[1], but the main use that my program focuses on is the game AI Dungeon 2.

AI Dungeon and AI Dungeon 2[2] is a game that uses the GPT-2 and GPT-3 language models to create stories. As mentioned, it uses the inputs that the user (or users for multiplayer) enters into the story, and it will attempt to create the next piece of the story. It's usually used in conjunction with other role-playing games like Dungeons and Dragons, but it can also be enjoyed on its own.

However, when a person goes back to reread the story that was created, it can be surprisingly human at times. The story beats that the AI generates are preexisting corpuses for the GPT-2 and GPT-3 language models to use, but the way in which the text is created is dependent on normal

writing conventions like grammar and punctuation, and on the way the users type. This creates plenty of stories that may sound like it was only created by one individual, when in reality, an AI has co-written the story told.

As the Generative Pre-Trained Transformer gets better and better with each iteration, it quickly becomes nearly impossible to discern as to which is human and AI created. So, I have decided to create a program that can attempt to differentiate between one and the other. However, I am going into this program with the knowledge that it may not actually be possible, which means that the Generative Pre-Trained Transformer is so effective at creating human writing that it is indistinguishable from even another program dedicated to tell the difference between one another.

Previous Research

While there were not too many pieces of research that particularly revolved around checking whether the identity of a writer was a human or an AI, there has been plenty of research for the language model, the GPT-2.

The paper closest to this project titled "The workweek is the best time to start a family – A Study of GPT-2 Based Claim Generation" used GPT-2 to create statements, and fact-checked the statements that were generated, both manually and via a program similar to the program for this paper. The scope of this particular project is far larger than the scope of the Human vs. AI project, in terms of what needs to be analyzed for the individual lines. For example, the source reads as follows: "Throughout this work, we consider debatable topics which correspond to a single Wikipedia title, phrased as a suggestion for a policy – e.g., We should increase the use of telemedicine, or as a valuation analysis – e.g.,telemedicine brings more harm than good."[3] In the end, my project prioritized the sentence structure instead of topics, but this paper is still relevant, should I choose to expand this project further.

Another piece of research is the paper "GenAug: Data Augmentation for Finetuning Text Generators." The project associated with it also utilized GPT-2 as well, but instead of using it to create statements and check for truth within the statements created, the project behind this particular paper instead used existing Yelp Reviews for alteration to examine the efficacy of the alterations that were made, and see if those alterations are viably understandable. According to the paper, "insertion of character-level synthetic noise and keyword replacement with hypernyms are effective augmentation methods. We also showed that the quality of generated text improves to a peak at approximately three times the amount of original training data."[4] This gave a great perspective as to how to proceed with the project. If AI Dungeon creates text solely through the GPT-2 model, understanding how the improvements can be made to existing text can aid the understanding as to how the AI would generate these texts in the first place. Alternatively, if these improvements are lacking in the text created by AI Dungeon, figuring out if text was AI generated or human generated might prove to be simpler than anticipated.

Moving towards the direction of the Machine Learning aspect of AI Dungeon, one paper titled "'What is Relevant in a Text Document?':An Interpretable Machine Learning Approach" offers an avenue towards analyzing the stories generated by AI Dungeon. The paper discusses how the developed program for this paper can annotate large documents into smaller ones by figuring out the important aspects of any given document. In particular, this may prove to be useful in parsing out the individual responses between AI and humans, for the actual corpus. According to the paper, "In the present work, we propose a method to identify which words in a text document are important to explain the category associated to it. The approach consists of using a ML classifier to predict the categories as accurately as possible, and in a second step, decompose the ML prediction onto the input domain, thus assigning to each word in the document a relevance score."[5] The project associated with this paper can help provide a good amount of insight towards figuring out what would be more aligned to be from an AI.

Lastly, the story element presents a unique challenge, one dilemma that wouldn't be present in analyzing other forms of documents, like legal documents or articles. The paper "Beyond Canonical Texts: A Computational Analysis of Fanfiction" discusses the challenges with stories as a whole, in the context of fan-made works versus the original source. It would act similarly to the human versus AI dilemma my project aims to address. In the paper, it read as follows: "The systematic ways in which fanfiction stories differ from their canonical works can give insight into the

characteristics of a story that are desired by fans but may be missing from the mainstream canon.We investigate [this] question: Is there a difference between the characters emphasized in fanfiction compared to the original canon?"[6] The project in this paper compared the original source to the fanwork, scanning for any useful differences. The project revealed what fans would want to have in their stories, and what fans would leave out from their stories.

Overall, these different pieces of research were helpful for understanding the challenges and the ideas behind my project, but they all were fairly peripheral to the main topic; whether the AI can be so great at writing, it would be seen as human, by another program designed specifically to determine the identity of the writer.

Methodology

To make a program that analyzes text to determine identity, we will need to use a Naive Bayers identifier. The Naive Bayers identifier will scan the documents (or in this case "actions") and, depending on previous results and a bit of prediction, it will try to determine the identity of the writer. There are five total classifications for the writing that will be used as features for the analysis of the "actions", which are provided by the library of the Natural Learning Toolkit, or NLTK. These features are punctuation, complexity, function words, lexical composition, and syntactic composition.

Punctuation is by far the most obvious feature. It identifies the punctuation of each action to determine a result. Though most people would be fairly hard-pressed to determine the identity of a writer based on punctuation and punctuation alone, it still provides a use, as the program can detect patterns that humans cannot.

Complexity is how complex the writing is. There are multiple factors for this aspect. There are the length of words, number of "long words" in a sentence, and the length of sentences. From personal experience, I can attest that the AI in AI Dungeon 2 usually does not utilize run on sentences, and instead relies on multiple short ones. This aspect, I hypothesize to be the most accurate feature to analyze the text.

Function Words are words that do not particularly have any meaning, but they do provide grammatical context and relationships with all the words in a sentence. The easiest example is pronouns. Pronouns provide a lot of context, but don't mean much on their own. The word "he" does not mean anything, aside from the subject being male. However, in the sentence, "Michael is in a house that he owns," the reader can understand that "he" refers to Michael.

The feature for lexical composition analyzes the most common words in each action. For both human and AI writers, this aspect of the actions that they input might be the most interesting one to keep track of. For a human writer, they might be compelled not be repetitious with their words, or else the sentences that they write may sound flat. For a similar reason, an AI may be trained to avoid this type of writing.

Syntactic composition checks the actual composition of the sentence itself. The tags that are available in the NLTK provide us with the types of word structures each writer prefers to use. From there, the program can determine who the writer is.

Of course, these individual features can have their own sets of accuracy. The accuracy of two features can be better, or even worse than the individual features. So, the code I have prepared will run through each combination of all five features, resulting in 31 different runthroughs of the existing corpus, which will be discussed further in the next section. The accuracy is determined based on the normalized, or Gaussian, distribution of the predictions, where it determines a prediction based on prior actions and the structures in the currently scanned action.

Data Collection

All the data comes from the AI Dungeon 2's Griffin Model. The Griffin Model utilizes GPT-2 for its generative text. As such, it is not the pinnacle of what can be created, but the text can be good enough to fool most people, and only after extensive study of how the text is created would someone be able to differentiate the text between human-generated and AI-generated. The Dragon Model utilizes GPT-3 for its generative text, but due to limited resources for this project, I had to stick with the Griffin model instead.

All the text that is used for this experiment comes from fellow friends, as to not entirely bias the experiment with my own personal writing. Normally, I would want to use the publicly available data from users across AI Dungeon's user base. This would allow much more variation

in the human styled writing, but as I mentioned, even the Griffin Model is capable of producing good enough text to be indiscernible to the average user, and AI Dungeon currently does not provide a venue to detect which passages within the text were AI generated. As such, all data had to be made, collected, and organized on my own.

The data collection was simple enough. Using a screen recording software, I had my friends contribute by having their sessions playing the game be recorded. AI Dungeon may not be able to save which inputs were created by the user or AI, but it can save the final story, and the text created while the game is played will be generated in real time. So, after each session was over, I combed through the footage and checked line by line which "actions" were AI generated and which "actions" were human generated. All of those are stored in .csv files within the corpus.

The story created by the AI also needed to be adjusted slightly. Nearly every line provided by a human includes a greater than sign (>), which the program can pick up. Since it uses the Naive Bayers classifier to determine the writer's identity, it proved to be incredibly problematic, as it skewed the results heavily, and the program was able to detect which writer wrote any given line with a 95% accuracy because of it. To rectify this issue, I removed those signs in the final corpus.

The corpus for the experiment is unfortunately smaller than my original plan. I intended to utilize 1000 "actions" for the experiment, and I was only able to garner about 400 due to limited resources. With this in mind, this did necessitate an alteration for the finalized experiment. Instead of relying on one runthrough for the program, the experiment will need to run the program three times, with an increasing corpus size. This will allow a trendline to be created, as the program runs through a larger and larger corpus.

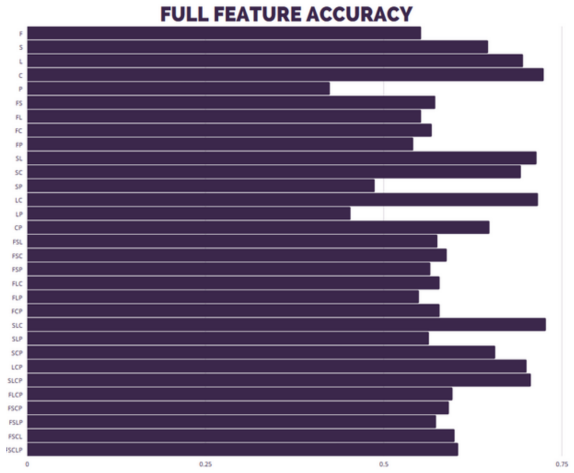


Fig 1

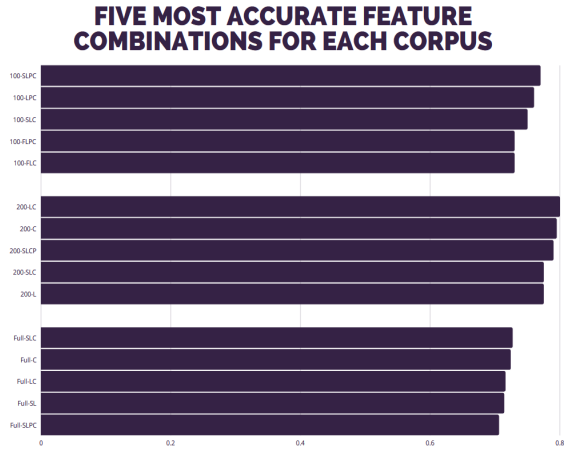
Feature	Accuracy(%)		Feature	Accuracy(%)
F	55.2%		FSC	58.8%
L	64.6%		FSP	56.5%
S	69.5%		FLC	57.8%
C	72.4%		FLP	54.9%
P	42.4%		FCP	57.8%
FS	57.2%		SLC	72.7%
FL	55.2%		SLP	56.3%
FC	56.7%		SCP	65.6%
FP	54.1%		LCP	70.0%
SL	71.4%		SLCP	70.6%
SC	69.2%		FLCP	59.6%
SP	48.7%		FSCP	69.1%
LC	71.6%		FSLP	57.3%
LP	45.3%		FSCL	59.9%
CP	64.8%		FSCLP	60.4%
FSL	57.5%			

Results

So, the first important question to ask is "how should we analyze each action?" We are given the five features, and there are a total of 31 combinations of each feature, so one of those combinations should be suitable for analyzing the data.

In Fig. 1, we can see the full analysis of the corpus, with each combination of features, with F standing for Function Words, S standing for syntax, L standing for lexical, P standing for punctuation, C standing for complexity, and a combination of those characters standing for a combination of those features. The number values at the bottom of the graph represent the percentage accuracy of each combination. In this analysis, we can determine a couple of pieces of information. First, a combination of all five features is NOT as effective as only some of the features. This is important, as future experiments may not need to utilize all five features, though

there is something more significant with that in Fig. 2. Second, analyzing each feature respective to every other feature showcases that the most accurate ways to differentiate between a human writer and an AI writer uses the lexical, syntax, and complexity features. Punctuation poses a mild hindrance, and the functional words feature poses a much heavier hindrance on accuracy.



Corpus Size	Feature	Accuracy(%)
100	SLCP	77.0%
100	LCP	76.0%
100	SLC	75.0%
100	FLPC	73.0%
100	FLC	73.0%
200	LC	80.0%
200	C	79.5%
200	SLCP	79.0%
200	SLC	77.5%
200	L	77.5%
384	SLC	72.7%
384	C	72.4%
384	LC	71.6%
384	SL	71.4%
384	SLPC	70.6%

Of course, the next logical step is seeing how the program handles different sized corpuses. In Fig. 2, it showcases that the accuracy for determining if the actions are written by AI or human is at worst 73%, which means that the program is capable of correctly detecting the identity of the writer 73 times out of one hundred actions, which is far better than initially anticipated. However, I do understand that the corpus is smaller than ideal, and the accuracy may decrease with a far larger corpus. However, given that it has a corpus of roughly 400, I would like to believe that the decrease in accuracy would not reach below 60%.

Interestingly enough with Fig. 2, the accuracy increased with the size-200 corpus. If I had to hazard a guess, the initial size-100 corpus was simply not large enough to learn the patterns of human and AI, but the patterns were more clear with the extra 100 items in the size-200 corpus. The drop afterwards for the full corpus is far more representative of the program. I hypothesize that because of the GPT-2 language model's adaptability towards the writer's style, differentiating between the two became more difficult for the program.

Conclusion

A question that I addressed in the presentation is this: Why this project? There's a couple of motivations behind wanting to discover if AI has reached the point to being able to successfully mimic human writing. Firstly, automated writing is simply more common. As mentioned earlier, it is commonly used for newspaper articles, but its use doesn't end there. AI Dungeon is a game that uses language models to generate text. With the way AI Dungeon works, it makes sense to see that idea extend to novels, as it simplifies the writing process.

And that extends to the many ethical dilemmas automated text can yield. Plagiarism would be difficult to identify with text that was generated and original. Spams and phishes would become more effective in their job, which is undeniably bad. A language model could be combined with misinformation bots that flood social media with misleading and extreme propaganda, adding a sense of unease and panic at the cost of those bots, which can be done in the thousands. Even the matter of needing to credit an AI for co-written articles may prove to be an issue, as it may be necessary to credit the very tool that helped create the article.

From the project, it's safe to say that even though it is fairly effective at detecting the differences between humans and AI, it isn't perfect. With at worst a 73% chance and at best an 80% chance of correctly deducing the identity of any given work, it showcases that it's already difficult for even another program dedicated to differentiate the robotic and organic authors. It may be partially effective at curbing the issues that were

mentioned previously, but that efficacy will only trend downwards as these language models get better and better.

Future Work

For future work, there are many improvements with this particular project that I would like to tackle. For one, being able to work with more facets of the project would be an incredible help. A larger corpus consisting of more varied writing of both human and AI, the Dragon (or GPT-3) model, and a differing approach are all different ways that this project can be taken, so much so that it may invoke a new project for each of these aspects. This project can be viewed as the "future work" of a different project worked on in the class, except with female and male writers instead of AI and human writers. The project code for this one is quite similar to the code of the previous experiment, but there were some necessary changes to more fit the needs of this project, such as the multiple runthroughs to detect which ways would be the best way to analyze the writings.

One worthwhile venture to simplify future work is simply contributing to AI Dungeon 2. AI Dungeon 2 is open sourced, and it is publicly available on Github. For a future project, incorporating a modified version of AI Dungeon 2 would be fantastic, even if it is as simple as tracking which actions resulted from which writer.

Beyond the project, it does open up a lot of work towards working with these language models. As amazing as they are, they can always be better. Although I had expectations that this project would showcase that it would be equally as difficult for a program to differentiate between human and AI, it would be interesting if the GPT-3 was capable of achieving the very antithesis of this project.

References

[1]: GPT-3. "A Robot Wrote This Entire Article. Are You Scared Yet, Human? | GPT-3." The Guardian, Guardian News and Media, 8 Sept. 2020, www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3.

[2]: Latitude. AI Dungeon. 10 Dec. 2020, https://aidungeon.io

[3]: Shai Gretz, Yonatan Bilu, Edo Cohen-Karlik, Noam Slonim. The workweek is the best time to start a family – A Study of GPT-2 Based Claim Generation. Accessed on 11/7/20. Accessed at

https://paperswithcode.com/paper/the-workweek-is-the-best-time-to-start-a.

[4]: Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, Eduard Hovy. GenAug: Data Augmentation for Finetuning Text Generators.Accessed on 11/7/20. Accessed at

https://paperswithcode.com/paper/genaug-data-augmentation-for-finetuning-text

[5]: Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, Wojciech Samek. "What is Relevant in a Text Document?": An Interpretable Machine Learning Approach.

Accessed on 11/7/20. Accessed at

https://paperswithcode.com/paper/what-is-relevant-in-a-text-document-an

[6]: Smitha Milli, David Bamman. Beyond Canonical Texts: A Computational Analysis of Fanfiction. Accessed on 11/7/20. Accessed at

https://paperswithcode.com/paper/beyond-canonical-texts-a-computational

Work Distribution

- Project Proposal: Yizhang Xie and Ryan Bautista
- Data Collection: Ryan Bautista
- Coding and Code Evaluation: Ryan Bautista
- Literature Review: Ryan Bautista
- Presentation: Ryan Bautista
- Research Paper: Ryan Bautista