Real or Fake Text?: Investigating Human Ability to Detect Boundaries between Human-Written and Machine-Generated Text

Liam Dugan[*], Daphne Ippolito[*], Arun Kirubarajan, Sherry Shi, Chris Callison-Burch

University of Pennsylvania
{ldugan, daphnei, kiruba, shershi, ccb}@seas.upenn.edu

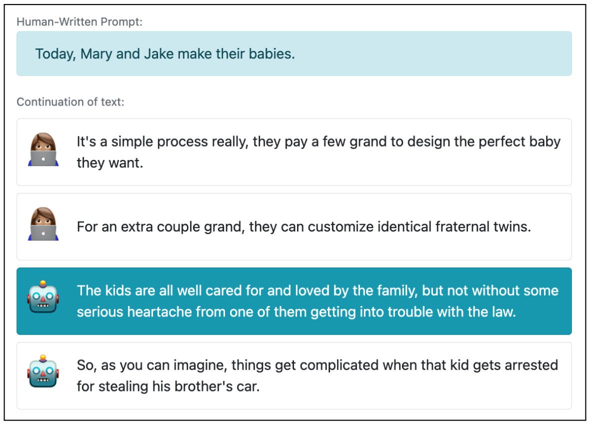
Abstract

As text generated by large language models proliferates, it becomes vital to understand how humans engage with such text, and whether or not they are able to detect when the text they are reading did not originate with a human writer. Prior work on human detection of generated text focuses on the case where an entire passage is either human-written or machine-generated. In this paper, we study a more realistic setting where text begins as human-written and transitions to being generated by state-of-the-art neural language models. We show that, while annotators often struggle at this task, there is substantial variance in annotator skill and that given proper incentives, annotators can improve at this task over time. Furthermore, we conduct a detailed comparison study and analyze how a variety of variables (model size, decoding strategy, fine-tuning, prompt genre, etc.) affect human detection performance. Finally, we collect error annotations from our participants and use them to show that certain textual genres influence models to make different types of errors and that certain sentence-level features correlate highly with annotator selection. We release the RoFT dataset: a collection of over 21,000 human annotations paired with error classifications to encourage future work in human detection and evaluation of generated text.

1 Introduction

Neural language models (LMs) are capable of generating increasingly natural-sounding text. One growing worry is that bad actors may attempt to pass off automatically-generated text as genuine. For example, Zellers et al. (2019) discuss the dangers of machine-generated news articles, Martens and Maalej (2019) document how easy it is to buy fake app store reviews, and Weidinger et al. (2021) chronicles how LMs can potentially be used to spread misinformation, fraud, and other harmful text. These harms will inevitably become more and more prevalent as language models become better and cheaper to deploy. Thus, it is important to answer the question: just how susceptible are humans to being duped by machine-generated text?

Existing studies of the ability of humans to detect generated text have focused on the binary question of whether or not a provided document contains any generated text at all (Ippolito et al. 2020; Gehrmann, Strobelt, and Rush 2019; Clark et al. 2021). In this work, we instead frame detection as a boundary-detection task: given a document that starts off as human-written and at some point transitions to machine-generated, can annotators detect the transition point? The boundary detection setting is more informative than the classification setting because it better aligns with how LMs are used to generate text in practice—in typical usage, a generative system is provided with a prompt and asked to produce a continuation. By measuring human skill at the boundary detection task, we can make progress toward quantifying the risks associated with LMs; while simultaneously evaluating the performance of different generative systems.



Today, Mary and Jake make their babies.

🧑‍💻 It's a simple process really, they pay a few grand to design the perfect baby they want.

🧑‍💻 For an extra couple grand, they can customize identical fraternal twins.

🤖 The kids are all well cared for and loved by the family, but not without some serious heartache from one of them getting into trouble with the law.

🤖 So, as you can imagine, things get complicated when that kid gets arrested for stealing his brother's car.

Figure 1: In the boundary detection task, players see one sentence at a time from a passage and try to guess when the passage transitions from human-written to machine-generated.

We collect annotations using RoFT, the website introduced in Dugan et al. (2020) which poses the boundary detection task as a game. In each game round, players are shown one sentence at a time and earn points for guessing close to the true boundary (Figure 1). They are also asked to select a reason for why they made their decision. In our analysis of these annotations, we find that players vary substantially in their detection ability and that factors such as the amount of time taken to complete a game round and total number of game rounds played correlate with success.

We discuss how various factors such as the genre of

[*] Equal Contribution

prompt (news, stories, etc.), the size of the model, and decoding strategy used affect generation quality. Finally, we examine the trends and errors which distinguish real from generated text and look at whether our players could pick up on these trends. In addition to producing valuable data for analyzing detectability, our study serves as the first large-scale attempt at using a gamified platform to analyze the detectability of generated text. Such a platform is easily extensible to support answering additional research questions in the future. All generations and annotations used in this paper are made publicly available to encourage further study of the detectability of machine-generated text[1].

2 Related Work

Previous research on understanding the ability of humans to detect machine generated text has mostly posed the task as a classification task—given a text example that is either entirely human-written or entirely machine-generated (aside from an initial prompt), annotators must predict whether it is human-written or machine-generated. On this task, Ippolito et al. (2020) reported that trained evaluators were able to achieve an accuracy of at best 71.4%, using generations from GPT-2 Large (Radford et al. 2019). In a follow-up study, Clark et al. (2021) demonstrated that annotators are able to distinguish GPT-2 XL generations with at best 62% accuracy, but they perform no better than random chance on GPT-3 (Brown et al. 2020) outputs.

Even after training evaluators to improve their detection abilities, detection accuracy on GPT-3 was only able to converge to around 55% (Clark et al. 2021). A study by Brown et al. (2020) reported similarly low performance (52%) on the detection of machine-generated news articles. Most recently, Ethayarajh and Jurafsky (2022) report that annotators rate GPT-3 outputs as significantly more human-like than human-written text itself, suggesting a detection accuracy of *worse* than random chance, underscoring a need to re-think our human evaluations from both a metrics perspective and a task perspective.

In this vein, another related area of research is asking annotators to explain why they think generated text is generated. He et al. (2021) created a dataset of generated text annotated with the errors that humans found in it, and Dou et al. (2021) proposed an error annotation schema for generated text. The errors we allow players to report in our experiments were inspired by these schemata.

Our work aims to address this need to diversify our evaluation task by using the RoFT detection game format introduced by Dugan et al. (2020) to analyze human detection performance across a variety of genres and generative systems. A similar detection task, where annotators guess whether turns in a conversation were generated, was posed by Deriu et al. (2020) as a way to evaluate dialog systems, but it has yet to be applied to language models more generally. We demonstrate the feasibility of applying such an evaluation framework to more general language models.

3 Experimental Design

3.1 The Real or Fake Text Game

Our study uses data collected through the "Real or Fake Text" (RoFT) annotation platform (Dugan et al. 2020). RoFT is a turn-based game where a player first selects a domain of text (news articles, recipes, short stories, or speeches). The player then plays a series of game rounds. In each round, the player is shown a starting sentence which they are told comes from a real human-written document. They are then shown subsequent sentences, one at a time. Each subsequent sentence may be the true continuation of the document, or it may be text generated by a language model. Once the sentences transition to being machine-generated, they will stay so for the rest of the 10-sentence passage.

After being shown each sentence, the player must decide whether they think that sentence was machine-generated or human-written. If the user selects "human-written," another sentence is displayed. If the player selects "machine-generated," the game round ends and the true author (machine or human) for each sentence is revealed. Before submitting their selection, the player is able to select a reason to explain their choice of sentence. Thus, the player's goal in RoFT is to correctly identify the sentence at which a passage transitions from being human written to being generated by a language model. We claim that this task formulation is more faithful to how generated text appears in real world scenarios, since generating with a prompt is the standard way to achieve controllability, and malicious actors will not reveal what portion of a generation is the human-written prompt.

3.2 Datasets

For our study we sampled prompts from four diverse genres of text. For each genre, we selected a corpus of text from the genre, sampled documents from the corpus, sentence-segmented them, and filtered out all documents with less than ten sentences. Then, for each document, we randomly select one of the ten sentences to be the end of the prompt and replace all following sentences with a machine generated continuation. This results in an even distribution over prompt lengths, with approximately 10% of examples being fully human-written. Our four genres of prompts are as follows:

News Articles. Documents were drawn from the New York Times Annotated Corpus (Sandhaus 2008), which contains 1.8 million articles published by the Times between 1987 and 2007. Our hypothesis was that this domain would be challenging for models since news requires factual accuracy, which state of the art models have been shown to struggle with (Nakano et al. 2021; Lin, Hilton, and Evans 2022).

Presidential Speeches. Documents were drawn from the presidential speech corpus (Brown 2016), which contains 963 speeches given by presidents of the United States, with dates ranging from 1789 to 2015. Our hypothesis was that the sort of first-person rhetoric found in these speeches would be easy for models to impersonate since political speech and first-person speech are plentiful in web-based training data.

[1] https://github.com/liamdugan/human-detection

Genre	# Avail	# Annotations Raw	Filter	Avg Ann/Cont.	Continuation Sources	Top-p Value
News	1,838	7,806	4,488	2.97	GPT-2 XL (80%) Human (14%) Baseline (6%)	1.0 (31%) 0.4 (29%) 0.0 (33%)
Stories	9,864	8,007	4,614	2.53	GPT-2 (58%) GPT-2 XL (22%) Human (18%)	1.0 (9%) 0.4 (82%) 0.0 (9%)
Recipes	7,258	17,978	7,709	2.13	Finetune (67%) GPT-2 XL (22%) Human (12%)	0.4 (100%)
Speeches	297	8,374	4,835	16.28	CTRL-Pol (54%) CTRL (34%) Human (12%)	0.4 (100%)

Table 1: Statistics on the game rounds available and annotations collected. Players were asked to play 50 rounds in an assigned genre, after which they could choose any genre. Recipes was the most popular genre. Top-p is the Nucleus Sampling parameter. "Finetune" is our fine-tuned GPT-2 XL recipe model and "CTRL-Pol" is CTRL using only the `[Politics]` control code (§5).

Stories. Fictional stories were drawn from the Reddit Writing Prompts dataset (Fan, Lewis, and Dauphin 2018), a corpus of amateur short stories scraped from the r/WritingPrompts subreddit[2]. We hypothesized that this domain would be easy for models since the writing quality of the stories is not especially high (which lowers the bar for the model generation quality), and factuality is not as important in a fictional domain.

Recipes. Recipes were extracted from the Recipe1M+ dataset (Marin et al. 2019). Recipes were parsed slightly differently than the other domains. We set the first "sentence" of each document as the name of the recipe and the ingredient list, and each subsequent "sentence" was a step in the recipe. Some recipe steps were more than one sentence. We hypothesized that this dataset would be difficult for models due to the closed-ended and structural nature of the task and the reliance on common sense.

3.3 Awarding Points

In each game round, the player is awarded points based on how close their selection was to the true boundary sentence (i.e. the first machine-generated sentence in the passage). Players were awarded 5 points for correctly choosing the boundary sentence and $\max(5-n, 0)$ points for a guess n sentences after the boundary. Players were not awarded points for guessing a sentence before the boundary. Players were able to see how many points they earned in each category on their profile page and compare their performance with fellow players on the leaderboard page. In the Findings section (Section 4), we report mean score earned as the predominant evaluation metric. This metric has high correlation with other more standard metrics (see Appendix for more detail).

Some students did discover that by always choosing the last sentence, they could game the system to earn points without putting in effort. We accounted for this tendency and filtered out all suspicious annotations (see Appendix).

3.4 Players

Players were recruited from two sections of an Artificial Intelligence course for graduate students and senior undergraduates at the University of Pennsylvania. Each participant was randomly assigned a single genre for their first 50 annotations, after which they were allowed to choose between genres. All data used in our study is fully anonymized and only collected from students who explicitly consented to having their annotations used for research purposes. Additionally, the ethics of this study were reviewed and approved by the University of Pennsylvania's Institutional Review Board.

We separated our participants into two groups. Group A was our control group. They were asked to play 30 minutes of the RoFT annotation game and received 2 points of class extra credit as compensation regardless of their score. The second section (Group B) was explicitly told they would be awarded $\min(p/250, 2)$ points of extra credit toward their final grade, where p was the total number of points the student earned across all rounds played. While Group A was only given basic instructions on how to play the game, Group B was given a detailed guide for how to identify generated text. Statistics on the number of annotations collected from each group can be found in the Appendix.

We note that university students taking an advanced artificial intelligence course are not reflective of the global population of English speakers, and the results presented in this paper may not reflect the general population's ability to detect machine-generated text. However, we believe our set of students are sufficient for a preliminary study and we would like to see broader representation in future work.

3.5 Continuation Sources

One of the main goals of our study is to investigate how specific model attributes (size, sampling strategy, etc.) affect our players' ability to detect generated text. In order to do this we structure our continuation sources as follows: We first decide on a base model configuration, then for each experiment we vary exactly one aspect of that base model and observe how human performance changes. We generally use only one (sometimes two) genres per comparison due to budget constraints but encourage future work to address this limitation.

For our base continuation model we use pre-trained GPT-2 XL with nucleus sampling parameter of $p = 0.4$ (Holtzman et al. 2020) and repetition penalty of 1.2 (Keskar et al. 2019). We generate continuations using this model on News, Stories, and Recipes to serve as our base for comparison. As an additional sanity check on annotator skill, we also include 100 game rounds in the News domain where instead of transitioning to an LM-generated continuation, the passage transitions to a completely different news article selected at random. We expected these game rounds to be trivial for players.

For our model comparisons, we first generate continuations on Stories with GPT-2 small in order to investigate how model size affects detectability. Then, we investigate whether

[2] https://www.reddit.com/r/WritingPrompts/

or not noisy text is easier to detect by generating continuations on News and Stories with GPT-2 XL using $p = 0.0$ (argmax sampling) and $p = 1.0$ (random sampling). In our third comparison, we fine-tune GPT-2 XL on Recipes and compare the output to our pre-trained base model. Finally, for the last comparison we look at the effects of topic control codes by generating continuations on Speeches using the CTRL model (Keskar et al. 2019) rather than GPT-2.

CTRL is a 1.6B parameter LM that allows users to pass in one of 50 pre-defined control codes to condition the model to generate in a particular style. For half of the generations, we used the "[Politics]" control code while for the other half we randomly selected a control code each time. We hypothesized that presidential speeches would be a good domain for this comparison because speeches have a very distinctive formal style and are typically very semantically similar.

We report the results of our main study of detectability in Section 4. We report the results of our comparison experiments in Section 5. Finally, we report extra RoFT-Specific analysis on topics such as model errors in Section 6.

3.6 Final Annotations Used for Analysis

In total, we collected 42,165 annotations over 7,895 different game rounds. After filtering for an even distribution over prompt length and removing players who exploited vulnerabilities (see Appendix), we ended up with a final dataset of 21,646 annotations over 7,257 continuations.

Table 1 gives a detailed breakdown of the dataset across genres and generation systems. For News, Stories, and Recipes, we had on average 2 players per continuation, while for Speeches, a smaller dataset, we had 16.

4 Main Findings

In this section we report the findings of our main detectability study. We analyze the accuracy of our players, their agreement, their improvement over time, the distribution of skill across players, and if said skill varies across genre. Error bars on all figures are 95% confidence intervals and the exact values and confidence intervals for all figures can be found in the Appendix.

Can humans detect generated text? We found that players were significantly better than random chance at the boundary detection task, correctly selecting the boundary sentence 23.4% of the time (chance being 10%). For rounds with at least one generated sentence, players selected a generated sentence as the boundary sentence 72.3% of the time.

The average number of points (§3.3) received per round by our players was 2.08, also well above random chance (1.31 points). Additionally, out of the 214 annotations we collected for our "sanity check" baseline, the mean score was 2.75, significantly higher than any of the true LM-backed systems (but still far from a perfect 5.00). For the remaining analyses, we will use average points earned per round ("mean score") as the primary measure of detection ability instead of accuracy. This measure correlates with accuracy as well as other metrics that one might consider using (see Appendix).

To measure inter-annotator agreement we use the Krippendorff's alpha co-efficient. This statistic measures how much disagreement there is between players compared to the amount one would expect by chance. Two players are considered to have agreed if they both guessed "machine-generated" on any sentence on or after the true boundary or if they both guessed the entire passage was human-written. Over all annotations, we found α=0.25, indicating only slight agreement amongst most players. However, among our top 10% of players (measured by mean score), there was high agreement, with α=0.40, suggesting that good players made similar errors.

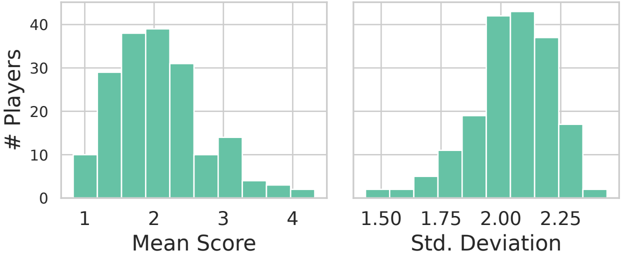


Figure 2: Histogram of mean score and standard deviation of score among players who completed at least 20 rounds. We see large gaps in skill between players, with some having significantly higher mean score and lower variance than others.

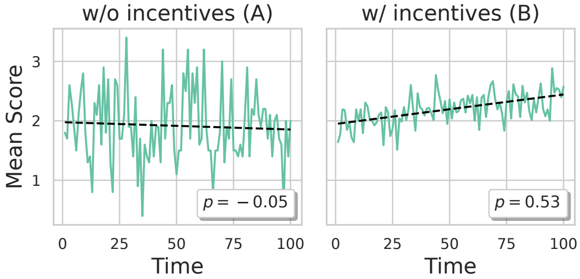


Figure 3: Performance over time for the two player groups (§3.4). Players in Group B, who were given extra instruction and incentives, improved over time ($\rho = 0.53$) while those in Group A did not ($\rho = -0.05$).

How much does player ability vary? We observed a large variance in the skill of individual players. In Figure 2 we report the distribution of mean scores and standard deviations of all players that completed more than 20 rounds. We see that some players have significantly higher mean score than others and we find that higher mean score also tends to coincide with lower standard deviation. Among players with mean score over 3.8, the average standard deviation was 1.84.

We also found that under the right conditions, players can exhibit improvement over time. In Figure 3 we report the performance over time across our two player groups. While we saw no correlation between number of rounds played and player score for our control (Group A), Group B, who were given extra credit proportional to their game score and extra instruction, did show improvement along with lower variance over time.

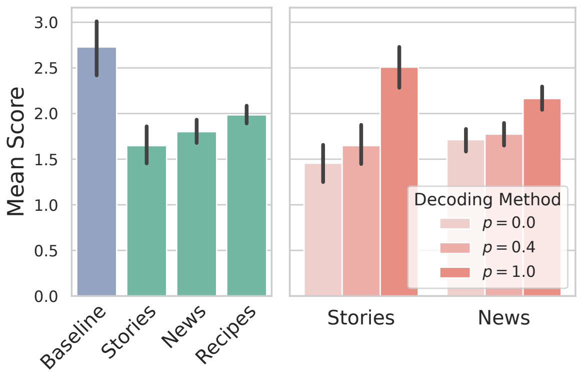


Figure 4: (left) Comparison of mean player score across different genres with GPT-2 XL p=0.4 against our "sanity-check" baseline (§3.5). (right) Comparison of mean player score across values of p for nucleus sampling (GPT-2 XL).

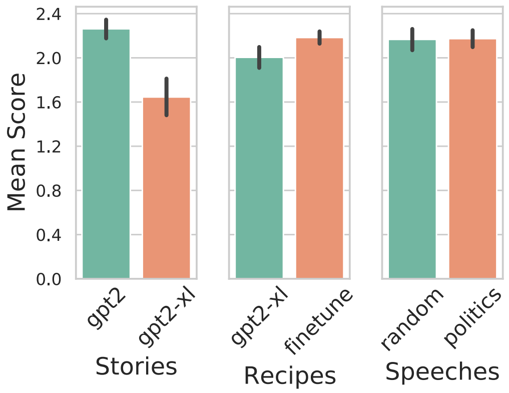


Figure 5: (left) For Stories, as model size increases (using p=0.4), detection becomes harder. (middle) For Recipes, extra finetuning does not significantly impact detectability. (right) For Speeches, using a "[Politics]" control has no impact on detectability.

Finally, among all questions asked on our exit survey, we found that the most predictive feature for annotator mean score was whether or not they reported that they had read our help guide[3]. This guide contained a taxonomy of model errors, a set of annotated examples, and other detection tips and common pitfalls. This finding, coupled with the previous findings on improvement over time and high skill variance, suggest that detection is a skill and that annotators can be trained to do well on the detection task.

Are some genres easier to detect? We found that certain genres were slightly easier than others. In particular, generated recipes were easier to detect than generated stories or news articles. Figure 4 (left) shows a comparison of mean player score across each genre.

For Recipes, we hypothesize that detection was made easier by the fact that the first human-written "sentence" in each game round was a semi-structured ingredients list, making it easy for players to check for contradictions—a step saying to mix in cream is probably generated if there is no cream ingredient. In addition, recipes often assume implicit unwritten knowledge which language models struggle to get right—a step saying to crack eggs cannot follow a step saying to whisk the eggs.

Indeed, if we look at the reasons given by our players as to why they selected certain recipes as generated (Figure 6), we see that continuations in the Recipes domain contain a much larger percentage of "common_sense" errors (26%) than those in the News (10%) or Stories (10%) domain.

5 Model Comparison Findings

In this section we report the results of our comparison experiments. These are one-off comparisons that investigate the effect of one particular variable on detection accuracy. Similar to Section 4, error bars on all figures are 95% confidence intervals and the exact values and confidence intervals for all figures can be found in the Appendix.

Does model size affect detection performance? Previous work has shown that language model performance scales with number of parameters (Kaplan et al. 2020; Hoffmann et al. 2022), so we expected players to be worse at detecting generations from larger models. Indeed, we found that players scored significantly higher when generations came from GPT-2 small (117M parameters) than when they came from GPT-2 XL (1.5B parameters).[4] In Figure 5 (left) we report the difference in mean player score between GPT-2 small and GPT-2 XL. The difference observed here is the most significant difference observed across all variables tested, reaffirming the correlation between scale and language model performance.

Are diverse generations easier to detect? Choice of decoding strategy is known to have significant impact on text quality (Zhang et al. 2021) and detectability (Ippolito et al. 2020). Choosing a lower value of p when generating with a nucleus sampling (Holtzman et al. 2020) decoding strategy produces less diverse but also less noisy text than choosing a higher value of p. In Figure 4 (right) we report our findings and see that players were significantly better at p=1.0 (pure random sampling) than the lower values, validating claims from earlier papers that LMs struggle to generate high-quality text with similar diversity to human-written text.

Does finetuning affect detectability? We had expected that finetuning on in-domain text would result in a model that was better able to fool humans. We report our results in Figure 5 (middle) and see that, counter to expectations, there was a small increase in player detection ability when generations came from GPT-2 finetuned on recipes compared with generations from pre-trained GPT-2. This is despite the

[3] http://github.com/liamdugan/human-detection/data/guide.pdf

[4] In an additional round of annotation, we also experimented with continuations from GPT-3 Davinci (175B parameters) but could not draw any statistically significant conclusions. We have included the results from that round of annotation in our Appendix.

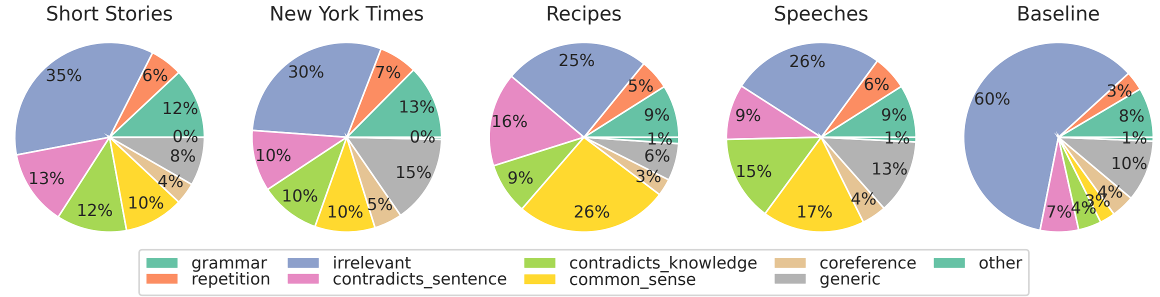


Figure 6: The reasons given by our players as to why they thought a given sentence was machine generated. Continuations for Stories, News, and Recipes were generated by GPT-2 XL with $p = 0.4$. Baseline refers to our sanity check baseline (§3.5). We see that GPT-2 XL tends to make more "common_sense" errors on recipes, more "irrelevant" errors on stories, and more "generic" errors on news. This is consistent with our intuition about what makes each of these domains challenging.

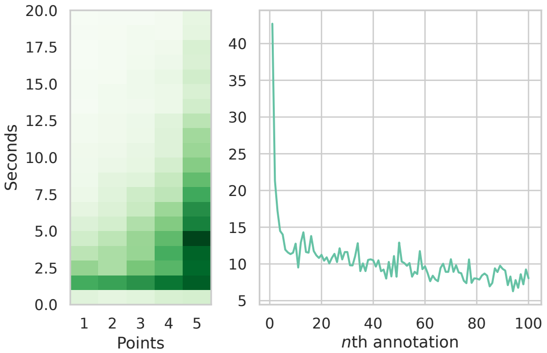


Figure 7: (left) Histogram showing the time taken per annotation and the score received. Scores tended to be higher when more time was taken. (right) Plot of time taken per annotation over time. We see that players get faster at the task over time.

fact that the finetuned model had close to half the perplexity of the pre-trained model on a held out test set of 50,000 recipes (4.781 vs. 8.979). While we can only speculate as to the amount of recipe knowledge present in the pre-trained model (GPT-2's training data is not publicly available), it is possible the pre-trained model already contained enough understanding of recipe-like text that it was not critical to do the extra-finetuning. We speculate that finetuning may have had more impact in a specialized or jargon-laden domain (e.g. legal, medical).

Do control codes affect detectability? CTRL is a 1.6B parameter LM trained with controllability in mind. At inference time, one can pass in one of 50 pre-defined control codes, such as "[Politics]" or "[Horror]", to condition the model to generate in a particular style. We investigated the efficacy of these control codes on the genre of presidential speeches by using "[Politics]" for half of the generations and randomly selecting a control code for the remaining half. We decided to use the presidential speeches genre for this experiment due to its unique and distinctive style. We report our results in Figure 5 (right) and find that use of the politics control code did not significantly affect players' ability to distinguish real from fake text. This is not to say that control codes do not affect generation; however, it does suggest that the cues used by players to detect generations may not be related to stylistic details, at least not within the genre of political speeches. Further work is needed to investigate whether control codes could have influenced detectability in other more specialized domains (e.g. legal, medical).

6 RoFT-Specific Insights

In this section we use specific capabilities of the RoFT platform such as time-tracking and sentence-level error annotation to investigate additional research questions about how and why humans select certain sentences as generated.

How much time did game rounds take? To understand how much time game rounds took, we logged the amount of seconds players spent on each sentence decision. We controlled for instances of players leaving a game open mid-annotation by applying $\min(120, t)$ to all recorded times t. We computed total time per annotation by summing the times for each sentence-level decision. In Figure 7 we report the time taken per annotation (left) and the time taken per annotation over time (right). Unsurprisingly, when players took longer on annotations, we found that they ended up receiving more points. We also found that players gradually got faster over time. Interestingly, while one might expect longer sentences to take more time to read and make decisions on, we found no correlation between time taken and length of sentence (ρ=-0.10). This indicates that players take time to think about the task beyond just reading the sentence.

What errors do humans look for when detecting generated text? Each time a player specified a sentence was machine-generated, they were asked to specify why they made this decision, selecting from a set of pre-defined options or otherwise writing down a custom reason. Figure 6 shows the distribution of these reasons across textual genres. For the Recipes domain we see a much higher proportion of "common_sense," errors, in News we see models generating

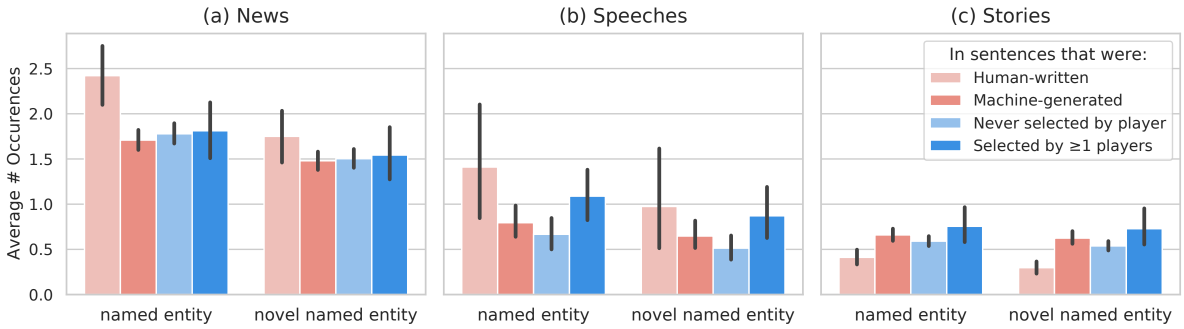


Figure 8: We see that human sentences tended to have a different number of named entities than generated sentences. Players picked up on the correct trend in Stories, but not in News or Speeches.

Reason	*n*	Mean Score
common_sense	2,432	2.566 ± 0.086
irrelevant	4,259	2.530 ± 0.064
contradicts_sentence	1,606	2.527 ± 0.105
contradicts_knowledge	1,411	2.262 ± 0.111
coreference	542	2.249 ± 0.176
repetition	728	2.128 ± 0.154
other	75	2.040 ± 0.483
generic	1,546	1.920 ± 0.101
grammar	1,539	1.780 ± 0.105

Table 2: The number of times each reason was given for text being machine-generated, and the mean score over those annotations. We see that when players select reasons like “grammar” or “generic,” they are much less likely to be correct than when selecting “common_sense” or “irrelevant.”

much more “generic” text, and in Stories we see more “irrelevant” text.

Table 2 shows, for each reason, the “reliability” of the reason, i.e. the average number of points earned on rounds where that reason was given. Like Clark et al. (2021), we see that conditioning on bad grammar is by far the least reliable way to detect generated text with a mean score of 1.78. Interestingly, the three most common reasons given (“common_sense,” “irrelevant,” and “contradicts_sentence”) were also the three most reliable, indicating that improving these attributes will lead to the biggest improvements in generation performance (and ability to fool humans).

What sentence-level features could be used to detect generated text? It has been well-studied how generated text differs in basic, measurable ways from human-written text, often due to the choice of decoding strategy. Given our task format, we wanted to measure how sentence length, part-of-speech distribution, and presence of novel named entities differed between the generated and human-written sentences in our dataset, and whether players were able to pick up on these differences. For this experiment we found surprisingly different trends across different genres. Figure 8 shows the results for named entities, where novel named entities are ones which occurred in the current sentence but not in any previous sentences. On News and Speeches, the generated sentences contained fewer named entities than the human-written sentences, while for Stories they had more. In the News domain, the sentences selected by players tended to have about the same number of entities as the ones not selected while in both Speeches and Stories, the sentences selected by annotators had slightly more named entities.

7 Discussion and Future Work

In this paper, we demonstrate the viability of the boundary detection task as a framework for soliciting human evaluation of natural-language generation systems. We conducted the largest study of generated text detectability to date and, in the process, replicated many previous results in the field, such as the improved performance of bigger models (Kaplan et al. 2020), the importance of decoding strategy selection (Ippolito et al. 2020), and the importance of incentivizing annotators (Clark et al. 2021).

In addition, we have provided new insights into the ways in which humans interact with generated text. We have shown that certain textual genres influence models to make different types of errors, that annotators can improve at the detection task over time, and that certain sentence-level features correlate highly with annotator selection.

We expect these results and the released dataset of generations and annotations to be of broad use to those studying detection. However, there is clearly still much left to be done in this space. One worthwhile avenue for future work would be to study how well automatic systems perform at the detection tasks, and whether we can predict when generated text will be especially difficult for human annotators to recognize.

Future work can also seek to build off our study by testing a larger set of models, genres, and other experimental conditions (finetuning, topic control, decoding strategy, etc.) as well as testing other frameworks for incentivizing annotators. It is also worth looking at the cases where continuations do not happen exactly on a boundary between sentences, as this is a core limiting assumption of our work. We also believe that more investigation is needed into exactly what annotators are thinking when they make their decisions and how we can give annotators the right tools to explain their thought processes.

Acknowledgements

This research is based upon work supported in part by the DARPA KAIROS Program (contract FA8750-19-2-1004), the DARPA LwLL Program (contract FA8750-19-2-0201), the Office of the Director of National Intelligence (ODNI) via the IARPA HIATUS Program (contract 2022-22072200005), and the NSF (Award 1928631). Approved for Public Release, Distribution Unlimited. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, ODNI, IARPA, NSF, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

All human subjects research conducted was approved by the Institutional Review Board (IRB) at the University of Pennsylvania under protocol number 819967. The IRB determined that the proposal meets eligibility criteria for IRB review exemption authorized by 45 CFR 46.101, category 2.

We would like to thank Google for their generous support through grants of Google Cloud Platform credits to support our server costs. We also thank Roblox and Salesforce for their financial support through gifts to our research group at the University of Pennsylvania.

We would like to thank the members of the lab for their suggestions and feedback. In particular, Alyssa Hwang was very influential in shaping the current version of this paper. Her great suggestions made the writing much clearer and much more understandable.

Finally, we are very grateful to the students of the Fall 2021 course CIS521 / CIT596 "Artificial Intelligence" for their participation and invaluable contributions to this study. Without their cooperation, none of this research would have been possible.

References

Brown, D. W. 2016. Corpus of Presidential Speeches. http://www.thegrammarlab.com. Accessed: 2020-11-15.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Clark, E.; August, T.; Serrano, S.; Haduong, N.; Gururangan, S.; and Smith, N. A. 2021. All That's 'Human'Is Not Gold: Evaluating Human Evaluation of Generated Text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 7282–7296.

Deriu, J.; Tuggener, D.; von Däniken, P.; Campos, J. A.; Rodrigo, A.; Belkacem, T.; Soroa, A.; Agirre, E.; and Cieliebak, M. 2020. Spot The Bot: A Robust and Efficient Framework for the Evaluation of Conversational Dialogue Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3971–3984. Online: Association for Computational Linguistics.

Dou, Y.; Forbes, M.; Koncel-Kedziorski, R.; Smith, N. A.; and Choi, Y. 2021. Scarecrow: A framework for scrutinizing machine text. *arXiv preprint arXiv:2107.01294*.

Dugan, L.; Ippolito, D.; Kirubarajan, A.; and Callison-Burch, C. 2020. RoFT: A Tool for Evaluating Human Detection of Machine-Generated Text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 189–196. Online: Association for Computational Linguistics.

Ethayarajh, K.; and Jurafsky, D. 2022. The Authenticity Gap in Human Evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 6056–6070. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 889–898.

Gehrmann, S.; Strobelt, H.; and Rush, A. M. 2019. GLTR: Statistical Detection and Visualization of Generated Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 111–116.

He, J.; Peng, B.; Liao, Y.; Liu, Q.; and Xiong, D. 2021. TGEA: An Error-Annotated Dataset and Benchmark Tasks for TextGeneration from Pretrained Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 6012–6025. Online: Association for Computational Linguistics.

Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; Casas, D. d. L.; Hendricks, L. A.; Welbl, J.; Clark, A.; et al. 2022. Training Compute-Optimal Large Language Models. *arXiv preprint arXiv:2203.15556*.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2020. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ippolito, D.; Duckworth, D.; Callison-Burch, C.; and Eck, D. 2020. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1808–1822. Online: Association for Computational Linguistics.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling Laws for Neural Language Models. arXiv:2001.08361.

Keskar, N. S.; McCann, B.; Varshney, L. R.; Xiong, C.; and Socher, R. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3214–3252.

Marin, J.; Biswas, A.; Ofli, F.; Hynes, N.; Salvador, A.; Aytar, Y.; Weber, I.; and Torralba, A. 2019. Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *IEEE Trans. Pattern Anal. Mach. Intell.*

Martens, D.; and Maalej, W. 2019. Towards understanding and detecting fake reviews in app stores. *Empirical Software Engineering*, 24(6): 3316–3355.

Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; Jiang, X.; Cobbe, K.; Eloundou, T.; Krueger, G.; Button, K.; Knight, M.; Chess, B.; and Schulman, J. 2021. WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners. Published on first author's personal website and OpenAI Blog.

Sandhaus, E. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12): e26752.

Weidinger, L.; Mellor, J.; Rauh, M.; Griffin, C.; Uesato, J.; Huang, P.-S.; Cheng, M.; Glaese, M.; Balle, B.; Kasirzadeh, A.; Kenton, Z.; Brown, S.; Hawkins, W.; Stepleton, T.; Biles, C.; Birhane, A.; Haas, J.; Rimell, L.; Hendricks, L. A.; Isaac, W.; Legassick, S.; Irving, G.; and Gabriel, I. 2021. Ethical and social risks of harm from Language Models. *arXiv preprint arXiv:2112.04359*.

Zellers, R.; Holtzman, A.; Rashkin, H.; Bisk, Y.; Farhadi, A.; Roesner, F.; and Choi, Y. 2019. Defending Against Neural Fake News. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zhang, H.; Duckworth, D.; Ippolito, D.; and Neelakantan, A. 2021. Trading Off Diversity and Quality in Natural Language Generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, 25–33. Online: Association for Computational Linguistics.