

# GLTR: Statistical Detection and Visualization of Generated Text

**Sebastian Gehrmann**

Harvard SEAS

gehrmann@seas.harvard.edu

**Hendrik Strobelt**

IBM Research

MIT-IBM Watson AI lab

hendrik.strobelt@ibm.com

**Alexander M. Rush**

Harvard SEAS

srush@seas.harvard.edu

## Abstract

The rapid improvement of language models has raised the specter of abuse of text generation systems. This progress motivates the development of simple methods for detecting generated text that can be used by and explained to non-experts. We develop GLTR, a tool to support humans in detecting whether a text was generated by a model. GLTR applies a suite of baseline statistical methods that can detect generation artifacts across common sampling schemes. In a human-subjects study, we show that the annotation scheme provided by GLTR improves the human detection-rate of fake text from 54% to 72% without any prior training. GLTR is open-source and publicly deployed, and has already been widely used to detect generated outputs.

## 1 Introduction

The success of pretrained language models for natural language understanding (McCann et al., 2017; Devlin et al., 2018; Peters et al., 2018) has led to a race to train unprecedentedly large language models (Radford et al., 2019). These large language models have the potential to generate textual output that is indistinguishable from human-written text to a non-expert reader. That means that the advances in the development of large language models also lower the barrier for abuse.

Instances of malicious autonomously generated text at scale are rare but often high-profile, for instance when a simple generation system was used to create fake comments in opposition to net neutrality (Grimaldi, 2018). Other scenarios include the possibility of generating false articles (Wang, 2017) or misleading reviews (Fornaciari and Poerio, 2014). Forensic techniques will be necessary to detect this automatically generated text. These techniques should be accurate, but also easy to convey to non-experts and require little setup cost.

## Human-Written

The programme operates on a weekly elimination process to find the best all-around baker from the contestants, who are all amateurs.

## Generated

The first book I went through was The Cook's Book of New York City by Ed Mirvish. I've always loved Ed Mirvish's recipes and he's one of my favorite chefs.

Figure 1: The top-k overlay within GLTR. It is easy to distinguish sampled from written text. The real text is from the Wikipedia page of The Great British Bake Off, the fake from GPT-2 large with temperature 0.7.

In this work, we argue that simple statistical detection methods for generated/fake text can be applied within a visual tool to assist in detection. The underlying assumption is that systems over-generate from a limited subset of the true distribution of natural language, for which they have high confidence. In a white-box setting where we have access to the system distribution, this property can be detected by computing the model density of generated output and comparing it to human-generated text. We further hypothesize that these methods generalize to black-box scenarios, as long as the fake text follows a similar sampling assumption and is generated by a large language model.

We develop a visual tool, GLTR, that highlights text passages based on these metrics, as shown in Figure 1<sup>1</sup>. We conduct experiments to empirically test these metrics on a set of widely-used language models and show that real text uses a wider subset of the distribution under a model. This is noticeable especially when the model distribution is low-entropy and concentrates most

<sup>1</sup>Our tool is available at <http://gltr.io>. The code is provided at <https://github.com/HendrikStrobelt/detecting-fake-text>

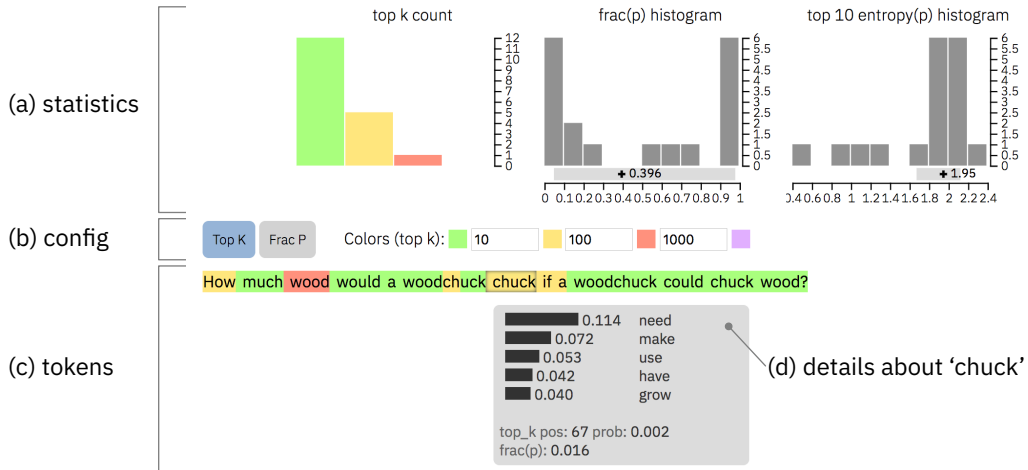


Figure 2: User interface for GLTR. On the top, we show three graphs with global information (a). Below the graphs, users can switch between two different annotations and customize the top-k thresholds (b). On the bottom, each token is shown with the associated annotation as heatmap (c). The tooltip (d) highlights information about the current prediction when hovering over the word “chuck”.

probability in a few words. We demonstrate in a human-subjects study that without the tool, subjects can differentiate between human- and model-generated text only 54% of the time. With our tool, subjects were able to detect fake text with an accuracy of over 72% without any prior training. By presenting this information visually, we also hope the tool teaches users to notice the artefacts of text generation systems.

## 2 Method

Consider the generation detection task as deciding whether a sequence of words  $\hat{X}_{1:N}$  have been written by a human or generated from a model. We do not have supervision for this task, and instead, want to use distributional properties of the underlying language. In the white-box case, we are also given full access to the language model distribution,  $p(X_i | X_{1:i-1})$ , that was used in generation. In the general case, we assume access to a different learned model of the same form. This approach can be contextualized in the evaluation framework proposed by Hashimoto et al. (2019) who find that human-written and generated text can be discriminated based on the model likelihood if the human acceptability is high.

The underlying assumption of our methods is that to generate natural looking text, most systems sample from the head of the distribution, e.g., through max sampling (Gu et al., 2017), k-max sampling (Fan et al., 2018), beam search (Chorowski and Jaitly, 2016; Shao et al.,

2017), temperature-modulated sampling (Dagan and Engelson, 1995), or even implicitly with rule-based templated approaches. These techniques are biased, but seem to be necessary for fluent output and are widely used. We therefore propose three simple tests, using a detection model, to assess whether text is generated in this way: **(Test 1)** the probability of the word, e.g.  $p_{\text{det}}(X_i = \hat{X}_i | X_{1:i-1})$ , **(Test 2)** the absolute rank of a word, e.g. rank in  $p_{\text{det}}(X_i | X_{1:i-1})$ , and **(Test 3)** the entropy of the predicted distribution, e.g.  $-\sum_w p_{\text{det}}(X_i = w | X_{1:i-1}) \log p_{\text{det}}(X_i = w | X_{1:i-1})$ . The first two test whether a generated word is sampled from the top of the distribution and the last tests whether the previously generated context is well-known to the detection system such that it is (overly) sure of its next prediction.

## 3 GLTR: Visualizing Outliers

We apply these tests within our tool GLTR (pronounced Glitter) – a Giant Language model Test Room. GLTR aims to both teach users what to be aware of when assessing whether a text is real, and to assist them in performing forensic analyses. It works on a per-instance basis for any textual input.

The backend supports multiple detection models. Our publicly deployed version uses both BERT (Devlin et al., 2018) and GPT-2 117M (Radford et al., 2019). Since GPT-2 117M is a standard left-to-right language model, we compute  $p_{\text{det}}(X_i | X_{1:i-1})$  at each position  $i$  in a text  $X$ . BERT is trained to predict a masked



Figure 3: On the left, we analyze a generated sample (a-c) with GLTR that is generated from a non-public GPT-2 model. The first sentence (a) is the prompt given to the model. We can observe that the generated text (b) is mostly highlighted in green and yellow, which strongly hints at a generated text. The histograms (c) show additional hints at the automatic generation. On the right, we show samples from a real NYT article (d) and a scientific abstract (e). Compared to the “unicorn” example, the fraction of red and purple words is much higher.

token, given a bidirectional context. Thus, we iteratively mask out each correct token  $\hat{X}_i$  and use a context of 30 words to each side as input to estimate  $p_{\text{det}}(X_i | X_{i-30 \dots i-1}, X_{i+1 \dots i+30})^2$ .

The central feature of the tool is the overlay function, shown in Figure 2c, which can render arbitrarily chosen top-k buckets (Test-2) as an annotation over the text. By default, a word that ranks within the top 10 is highlighted in green, top 100 in yellow, top 1,000 in red, and the rest in purple. GLTR also supports an overlay for Test-1 that highlights the probability of the chosen word in relation to the one that was assigned the highest probability. Since the two overlays provide evidence from two separate sources, their combination helps to form an informed assessment.

The top of the interface (Figure 2a), shows one graph for each of the three tests. The first one shows the distribution over the top-k buckets, the second the distribution over the values from the second overlay, and the third the distribution over the entropy values. For a more detailed analysis, hovering over a word (Figure 2d) shows a tooltip with the top 5 predictions, their probabilities, and the rank and probability of the following word.

The backend of GLTR is implemented in PyTorch and is designed to ensure extensibility. New detection models can be added by registering

<sup>2</sup>While BERT can handle inputs of length 512, we observed only minor differences between using the full and shortened contexts.

themselves with the API and providing a model and a tokenizer. This setup will allow the front-end of the tool to continue to be used as improved language models are released.

**Case Study** We demonstrate the functionality of GLTR by analyzing three samples from different sources, shown in Figure 3. The interface shows the results of detection analysis with GPT-2 117M. The first example is generated from GPT-2 1.5B. Here the example is conditioned on a seed text.<sup>3</sup> The analysis shows that not a single token in the generated text is highlighted in purple and very few in red. Most words are green or yellow, indicating high rank. Additionally, the second histogram shows a high fraction of high-probability choices. A final indicator is the regularity in the third histogram with a high fraction of low-entropy predictions and an almost linear increase in the frequency of high-entropy words.

In contrast, we show two human-written samples; one from a New York Times article and a scientific abstract (Figure 3d-e). There is a significantly higher fraction of red and purple (e.g. non-obvious) predictions compared to the generated example. The difference is also observable in the histograms where the fraction of low-probability words is higher and low-entropy contexts smaller.

<sup>3</sup>In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English

Feature	AUC
Bag of Words	$0.63 \pm 0.11$
(Test 1 - GPT-2) Average Probability	$0.71 \pm 0.25$
(Test 2 - GPT-2) Top-K Buckets	$0.87 \pm 0.07$
(Test 1 - BERT) Average Probability	$0.70 \pm 0.27$
(Test 2 - BERT) Top-K Buckets	$0.85 \pm 0.09$

Table 1: Cross-validated results of fake-text discriminators. Distributional information yield a higher informativeness than word-features in a logistic regression.

## 4 Empirical Validation

We validate the detection features by comparing 50 articles for each of 3 generated and 3 human data sources. The first two sources are documents sampled from *GPT-2 1.5B* (Radford et al., 2019). We use a random subset of their released examples that were generated (1) with a temperature of 0.7 and (2) truncated to the top 40 predictions. As alternative source of generated text, we take articles that were generated by the autonomous Washington Post *Heliograf* system, which covers local sports results and gubernatorial races. As human-written sources, we choose random paragraphs from the bAbI task children book corpus (CBT) (Hill et al., 2015), New York Times articles (NYT), and scientific abstracts from the journals nature and science (SA). To minimize overlap with the training set, we constrained the samples to publication dates past or close to the release of the GPT-2 models.

Our first model uses the average probability of each word in a document as single feature (Test 1) and the second one the distribution over four buckets (highlight colors in GLTR) of absolute ranks of predictions (Test 2). As a baseline we consider a logistic regression over a bag-of-words representation of each document. We cross-validate the results by training on each combination of four of the sources (two real/fake) and testing on the remaining two.

**Results** As Table 1 illustrates, the GLTR features lead to better separation than word-features, both with and without access to the true generating model. The classifier that uses ranking information learns that real text samples from the tail of the distribution more frequently. The odds ratio for a word outside the top 100 predictions is

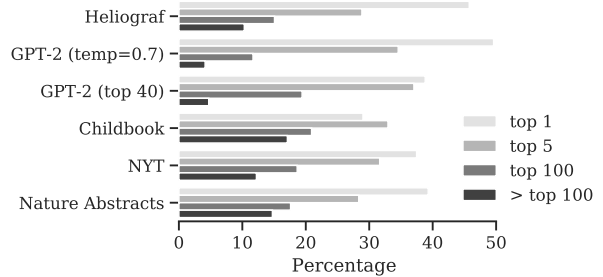


Figure 4: Distribution over the rankings of words in the predicted distributions from GPT-2. The real text in the bottom three examples has a consistently higher fraction of words from the tail of the distribution.

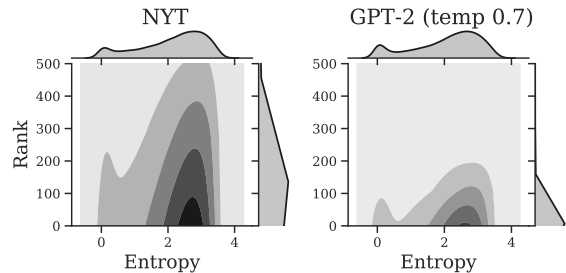


Figure 5: A kernel density estimate of the contextual entropy (Test 3) versus the next-word rank (Test 2) for NYT and GPT-2. Human-written text (NYT) is more likely to have high-rank words, even in low-entropy contexts.

5.32, while the odds ratio for being the top 1 prediction is 0.09. Figure 4 presents the distribution of rankings under GPT-2 and further corroborates this finding. Real texts use words outside of the top 100 predictions 2.41 times as frequently under GPT-2 (1.67 for BERT) as generated text, even compared to sampling with a lower temperature.

To get a better sense of how low-rank words enter into natural text, we look at the probability of each word compared to its relative rank. We hypothesize that human authors use low-rank words, even when the entropy is low, a property that sampling methods for generated text avoid. We compare the relationship of the entropy and rank of the next word by computing a Gaussian Kernel-density estimate over their distributions. As shown in Figure 5, human text uses high-rank words more frequently, regardless of the estimated entropy.

## 5 Human-Subjects Study

To evaluate the efficacy of the GLTR tool, we conducted a human-subjects study on 35 volunteer students in a college-level NLP class. Our goal was to both have students be able to tell generated

text from real, but also to see which parts raised the suspicion of the students. In two rounds, students were first shown five texts without overlay and then five texts with overlay and were asked to assess which texts were real within 90 seconds. In between the rounds, we presented a brief tutorial on the overlay and showed the example in Figure 1 but did not disclose any information about the study. For each participant and round, we presented two texts generated from GPT-2 with 0.7 temperature, one from Heliograf, and two from NYT.<sup>4</sup> We alleviated bias from the text selection by randomly assigning texts to either of the two rounds between students.

**Results** The results demonstrate the ease of use of the overlay. Without the interface, the participants achieved an accuracy of 54.2%, barely above random chance. While only 40% of texts were real, they trusted 56.0% of texts, Heliograf at a higher rate than GPT-2 (68.6% vs. 51.4%,  $p < 0.01$ ). The difficulty of the task without overlay was rated at 3.89 on a 5-point Likert scale, further supporting the need for assistive systems. With the interface, the performance improved to 72.3%. The average treatment effect shows an improvement of 18.1% with  $p < 0.001$ , even after controlling for whether a participant is a native speaker and how difficult they rated the task. 42.1% of the participants stated that the interface helped them be more accurate, and 37.1% found that it helped them to identify fakes faster.

**Qualitative Findings** The tool caused students to think about the properties of the fake text. While humans would vary expressions in real texts, models rarely generate synonyms or referring expressions for entities, which does not follow the theory of centering in discourse analysis (Grosz et al., 1995). An example of this is shown in the text in Figure 3b in which the model keeps generating the name Pérez and never refers to him as he. Another observation was that samples from Heliograf exhibit high parallelism in sentence structure. Since previous work has found that neural language models learn long linguistic structures as well, we imagine that sentence structure analysis can further be used for forensic analysis. We hope that automatic analysis and visualization like GLTR will help students better under-

<sup>4</sup>We randomly sampled one paragraph of text and resampled NYT if it was covering recent, well-known events.

stand the generation artifacts in current systems.

## 6 Related Work

While statistical detection methods have been applied in the past, the increase in language model power upends past assumptions in this area. Lavergne et al. (2008) introduce prediction entropy as an indicator of fake text. However, their findings are the opposite of ours (low entropy for generated text), a change which is indicative of language model improvements. Similar work finds that texts differ in perplexity under a language model (Beresneva, 2016), frequency of rare bigrams (Grechnikov et al., 2009), and n-gram frequencies (Badaskar et al., 2008). Similar methods that detect machine translation (Arase and Zhou, 2013). Hovy (2016) finds that a logistic regression model can detect generated product reviews at a higher rate than human judges, indicating that humans struggle with this task. Finally, we distinguish this task from detecting misinformation in text (e.g. Shu et al., 2017). We aim to understand the statistical signature and not the content of text.

## 7 Discussion and Conclusion

We show how detection models can be applied to analyze whether a text is automatically generated using only simple statistical properties. We apply the insights from the analysis to build GLTR, a tool that assists human readers and improves their ability to detect fake texts.

**Impact** GLTR aims to educate and raise awareness about generated text. To explain GLTR to non-NLP experts, we included a blog post on the web page with examples and an explanation of GLTR. Within the first month, GLTR had 30,000 page views for the demo and 21,000 for the blog. Numerous news websites and policy researchers reached out to discuss the ethical implications of language generation. The feedback from these discussions and in-person presentations helped us to refine our publicly released examples and explore the limits of our detection methods.

**Future Work** A core assumption of GLTR is that systems use biased sampling for generating text. One can imagine adversarial schemes that aim to fool our overlay; however, forcibly sampling from the tail decreases the coherence of a text which may make it harder to fool human readers. Another potential limitation are samples con-

ditioned on a hidden seed text. A conditional distribution will look different, even if we have access to the model. Our preliminary qualitative investigations with GLTR show a relatively short-range memory on this seed, but it is crucial to conduct more in-depth evaluations on the influence of conditions in future work. The findings further motivate future work on how to use our methods as part of autonomous classifiers to assist moderators on social media or review platforms.

## Acknowledgments

AMR gratefully acknowledges the support of NSF 1845664 and a Google research award.

## References

- Yuki Arase and Ming Zhou. 2013. Machine translation detection from monolingual web-text. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1597–1607.
- Sameer Badaskar, Sachin Agarwal, and Shilpa Arora. 2008. Identifying real or fake articles: Towards better language modeling. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.
- Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *International Conference on Applications of Natural Language to Information Systems*, pages 421–426. Springer.
- Jan Chorowski and Navdeep Jaitly. 2016. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*.
- Ido Dagan and Sean P Engelson. 1995. Selective sampling in natural language learning. In *Proceedings of the IJCAI Workshop on New Approaches to Learning for Natural Language Processing*, pages 41–48.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 889–898.
- Tommaso Fornaciari and Massimo Poesio. 2014. Identifying fake amazon reviews as learning from crowds. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 279–287.
- EA Grechnikov, GG Gusev, AA Kustarev, and A Raigorodsky. 2009. Detection of artificial texts. *RCDL2009 Proceedings. Petrozavodsk*, pages 306–308.
- James V. Grimaldi. 2018. [U.s. investigating fake comments on net neutrality](#). *The Wall Street Journal*.
- Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225.
- Jiatao Gu, Kyunghyun Cho, and Victor OK Li. 2017. Trainable greedy decoding for neural machine translation. *arXiv preprint arXiv:1702.02429*.
- Tatsunori B Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. *arXiv preprint arXiv:1904.02792*.
- Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children’s books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.
- Dirk Hovy. 2016. The enemy in your own camp: How well can we detect statistically-generated fake reviews—an adversarial study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 351–356.
- Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. *PAN*, 8:27–31.
- Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Louis Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. *arXiv preprint arXiv:1701.03185*.
- Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.
- William Yang Wang. 2017. ”liar, liar pants on fire”: A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.