# Generative Models are Unsupervised Predictors of Page Quality: A Colossal-Scale Study

Dara Bahri, Yi Tay, Che Zheng, Cliff Brunk, Donald Metzler, Andrew Tomkins

Google Research
Mountain View, CA, USA
{dbahri,yitay,chezheng,cliffbrunk,metzler,tomkins}@google.com

## ABSTRACT

Large generative language models such as GPT-2 are well-known for their ability to generate text as well as their utility in *supervised* downstream tasks via fine-tuning. Its prevalence on the web, however, is still not well understood - if we run GPT-2 detectors across the web, what will we find? Our work is twofold: firstly we demonstrate via human evaluation that classifiers trained to discriminate between human and machine-generated text emerge as *unsupervised* predictors of "page quality", able to detect low quality content without any training. This enables fast bootstrapping of quality indicators in a low-resource setting. Secondly, curious to understand the prevalence and nature of low quality pages in the wild, we conduct extensive qualitative and quantitative analysis over 500 million web articles, making this the largest-scale study ever conducted on the topic.

**ACM Reference Format:**
Dara Bahri, Yi Tay, Che Zheng, Cliff Brunk, Donald Metzler, Andrew Tomkins.

2021. Generative Models are Unsupervised Predictors of Page Quality: A Colossal-Scale Study. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441809

## 1 INTRODUCTION

The application of large neural language models for text generation has received a great deal of attention, from both the research community and the popular press [10, 18, 26–29, 38]. Many have raised concerns about the potential dangers of neural text generators in the wild, owing largely to their ability to produce human-looking text at scale.

Classifiers trained to discriminate between human and machine-generated text have recently been employed to monitor the presence of "deepfake" text on the web [29]. Little work, however, has been done in applying these classifiers for other uses, despite their attractive property of requiring no labels - only a corpus of human text and a generative model. In this work, we show through rigorous human evaluation that off-the-shelf human vs. machine discriminators serve as powerful classifiers of page quality. That is, texts that appear machine-generated tend to be incoherent or unintelligible. To understand the presence of low page quality in the wild, we apply the classifiers to a sample of half a billion English webpages. We analyze the results both qualitatively and quantitatively, providing breakdowns across dimensions such as time and topic distribution. We use two state-of-the-art methods for detecting machine-generated text: OpenAI's RoBERTa-based GPT-2 detector [20, 29] and GLTR [13]. These models are trained to distinguish GPT-2-generated text from human text. The goal of this work is not to improve detection modeling but to demonstrate the effectiveness of existing detection approaches on surfacing low quality pages on the web.

A webpage's quality is a function of many factors including but not limited to the reputability of the domain, its incoming or outgoing hyperlinks, the factual correctness of the content, the audio or video media present, and notions purely around the textual content. In this work we focus solely on *linguistic* or *language* quality (which we will define more precisely later) and we hereafter use the terms "page quality" and "language quality" interchangeably.

*Our Contributions.* The contributions of our work can be summarized as follows:

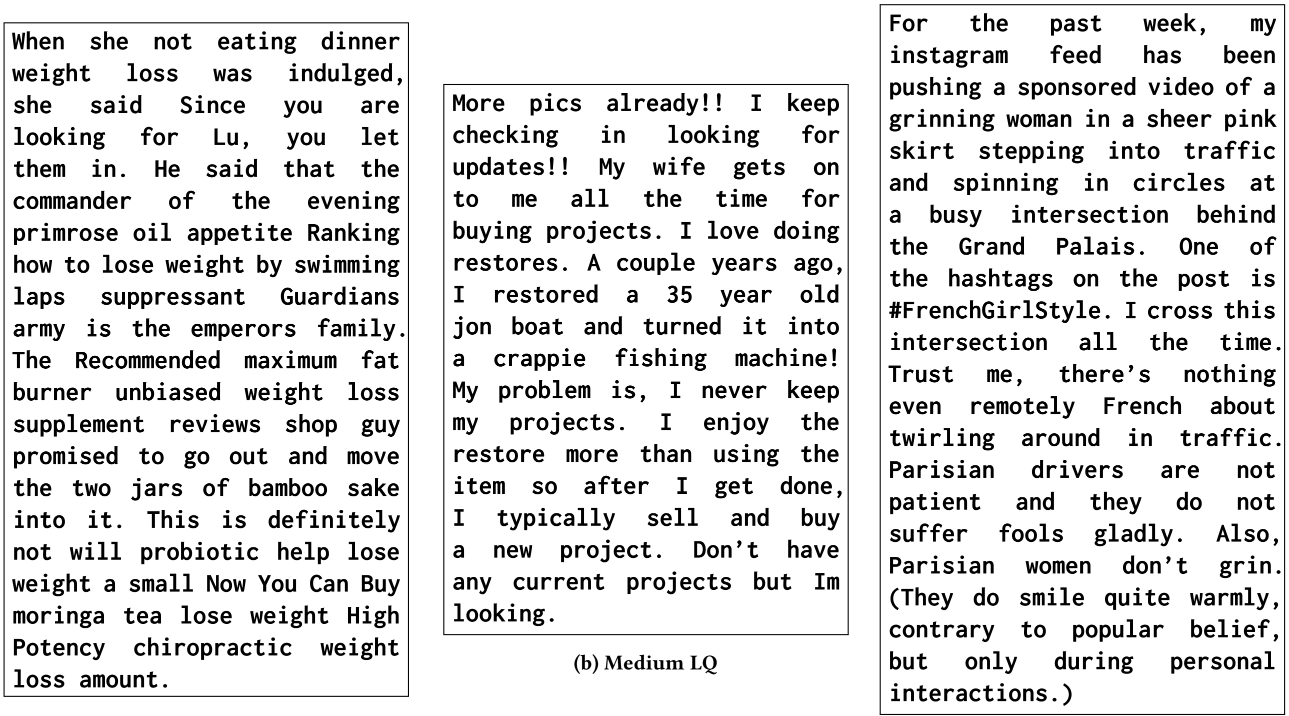
- We demonstrate through human evaluation that existing detectors of machine-generated text are effective at predicting low quality pages, outperforming, quite surprisingly, supervised spam classifiers. To our knowledge, this is the first use of machine detection for a different NLP task.
- Using half a billion webpages, we conduct the largest application of the detection models in the wild.
- We quantify the low quality pages that are surfaced by our detector models. We perform extensive analysis, breaking them down by attributes such as document length, age, and topic.
- We qualitatively characterize and categorize the nature of the low quality documents. We find traces of essay generation farms, machine translated text, keyword optimizations, and Not-Safe-For-Work (NSFW) content.

## 2 RELATED WORK

In this section, we briefly review work on text generation, human vs. machine detection, socially good and bad uses of neural generative models, and linguistic text quality.

*Generative Neural Language Models.* Neural text generation has attracted intense attention in recent years, largely owing to its ability to generate realistic text. This has been popularized by the GPT [26], GPT-2 [27], and GPT-3 [8] models, which demonstrate that

*WSDM '21, March 8–12, 2021, Virtual Event, Israel*
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8297-7/21/03.
https://doi.org/10.1145/3437963.3441809



When she not eating dinner weight loss was indulged, she said Since you are looking for Lu, you let them in. He said that the commander of the evening primrose oil appetite Ranking how to lose weight by swimming laps suppressant Guardians army is the emperors family. The Recommended maximum fat burner unbiased weight loss supplement reviews shop guy promised to go out and move the two jars of bamboo sake into it. This is definitely not will probiotic help lose weight a small Now You Can Buy moringa tea lose weight High Potency chiropractic weight loss amount.

**(a) Low LQ**

More pics already!! I keep checking in looking for updates!! My wife gets on to me all the time for buying projects. I love doing restores. A couple years ago, I restored a 35 year old jon boat and turned it into a crappie fishing machine! My problem is, I never keep my projects. I enjoy the restore more than using the item so after I get done, I typically sell and buy a new project. Don't have any current projects but Im looking.

**(b) Medium LQ**

For the past week, my instagram feed has been pushing a sponsored video of a grinning woman in a sheer pink skirt stepping into traffic and spinning in circles at a busy intersection behind the Grand Palais. One of the hashtags on the post is #FrenchGirlStyle. I cross this intersection all the time. Trust me, there's nothing even remotely French about twirling around in traffic. Parisian drivers are not patient and they do not suffer fools gladly. Also, Parisian women don't grin. (They do smile quite warmly, contrary to popular belief, but only during personal interactions.)

**(c) High LQ**

**Figure 1: Low, medium, and high language quality examples, as deemed by the GPT-2 detector and both human raters. Texts were minimally modified to protect the identity of the author.**

pre-training language models on large corpora enables not only superior downstream performance but can result in high quality text generation. Subsequently, [18] and [38] proposed CTRL and Grover respectively, which focused largely on text generation conditioned on article metadata. These models are auto-regressive and the sampling strategy used significantly affects generation quality [31]. Sampling methods include naive sampling from the full next-token softmax distribution, selecting the arg-max only, sampling from the top-$k$ scoring tokens [12] or the nucleus / top-$p$ head of the distribution [16], and various flavors of beam search [19, 32]. Top-$k$ and top-$p$ are commonly used in practice. Alternatives to auto-regressive models have been proposed.

*Detection Models.* Models detecting machine-generated text have also garnered interest in recent years. Early work in this area focused on feature-engineered models based on, for instance, $n$-gram or bag-of-words [5, 14]. Recent work has focused on leveraging pre-trained language models, e.g. by exploiting features obtained from language model outputs. GLTR [13] utilizes top-$k$ token rank features obtained from conditioning a language model on the input text. GLTR can also be used in a zero-shot setting where the probabilities assigned may be used as detectors without any training.

Generative models, such as Grover [35] and GPT-2, can also be used for sequence-level detection. This is in similar spirit to the standard fine-tuning methodology [11, 26]. Additionally, [6] proposes training an energy-based model that learns to rank a sequence of human text over the top-$k$ machine generations, primed on some amount of human text at either end of the sequence. Their model achieves strong results when train-test distributions are similar but struggles to generalize when there is a mismatch.

Notably, a recent study [31] proposes the new detection task of predicting the source of a generator model (i.e. the generator model architecture and hyper-parameter settings) from samples of generated text. The authors show that the task can be quite easy in certain cases.

*Misuse.* There are a number of potential ways that neural generative models can be misused, the most direct of which is in authoring content with malicious intent. OpenAI carried out a study to characterize the usage and detection of GPT-2 in the wild [29]. The study reports that their "threat monitoring did not find evidence of GPT-2 direct misuse in publicly-accessible forums but [they] did see evidence of discussion of misuse" and that "several governments have experimented with GPT-2 and other language models". While they provide some insight into how and by whom GPT-2 can be used, their findings are mostly qualitative and high-level in nature.

*Generation for Good.* Learning to detect machine text seems to imply that such text is generally considered to be bad or harmful. This is not always the case. Machine-generated text has a wide range of benevolent applications, like grammar correction [3] and neural machine translation [37]. Furthermore, there have been reports of positive GPT-2 usage, such as gaming [33], programming assistance [30], writing assistance [17], and poetry generation [34].

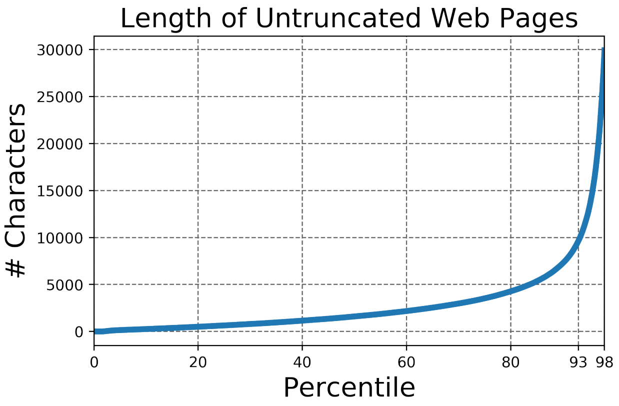


**Figure 2: Percentile plot of document length for Web500M. About 93% have fewer than 10k total characters, the length we truncate at.**

*Text quality.* We highlight some prior work on text quality, which can be roughly divided into two categories: (1) classifying the linguistic quality of human-written content, and (2) assessing and improving the quality and diversity of neural text generations. For human-written text, [4] characterizes the constituents of high editorial quality among news articles while [21] proposes a model that captures local coherence between sentences and demonstrates its use on readability assessment and essay scoring tasks. There have been significant efforts to catch low quality content, often referred to as spam, in large-scale web applications [7, 9, 22, 23]. For machine generations, BLEU [24] is a well-established method for measuring quality but it struggles at perceiving diversity, while Self-BLEU [39] captures diversity but struggles with quality. Constructing measures that capture both diversity and quality is an active area of research in the community [2].

# 3 EXPERIMENTS AND RESULTS

This section outlines our experimental setup and results.

## 3.1 Datasets

This section describes the datasets used in our experiments.

- **Web500M.** The core corpora used in our experiments consists of a random sample of 500 million English web documents obtained from the Common Crawl[1].
- **GPT-2-Output.** This public dataset (3) consists of the WebText test split and its GPT-2 generations under 8 different settings (2 sampling methods for each of 4 model sizes). The set is divided into train, test, and validation splits consisting of 250k, 5k, and 5k examples respectively.
- **Grover-Output.** We generated 1.2 million articles using pre-trained Grover-Base with a diverse range of sampling hyperparameters: top-$k$ sampling with $k$ ranging from 10 to 100 in steps of 10 and top-$p$ sampling with $p$ ranging from 0.65 to 0.90 in steps of 0.05. The model was conditioned on only the title field of articles from the publicly available CNN/DailyMail dataset [15]. We used the public Grover source code[2] for generating this data.

In order to bound document detection times, we truncate each document in all datasets to its first 10k characters. As depicted in Figure 2, about 93% of Web500M documents have fewer than 10k characters and are thus unaffected by the truncation.

## 3.2 Detectors

Our experiments utilize two recent detection methods, which we implement using the Tensorflow API of HuggingFace's Transformers library [1, 36].

To understand the significance of our two detectors as unsupervised predictors of page quality, we compare against a baseline that was trained explicitly on the spam classification task using the Scikit-Learn [25] python package. Since page quality and "spaminess" are strongly related, we expect that a classifier trained to detect the latter will transfer well for assessing the former.

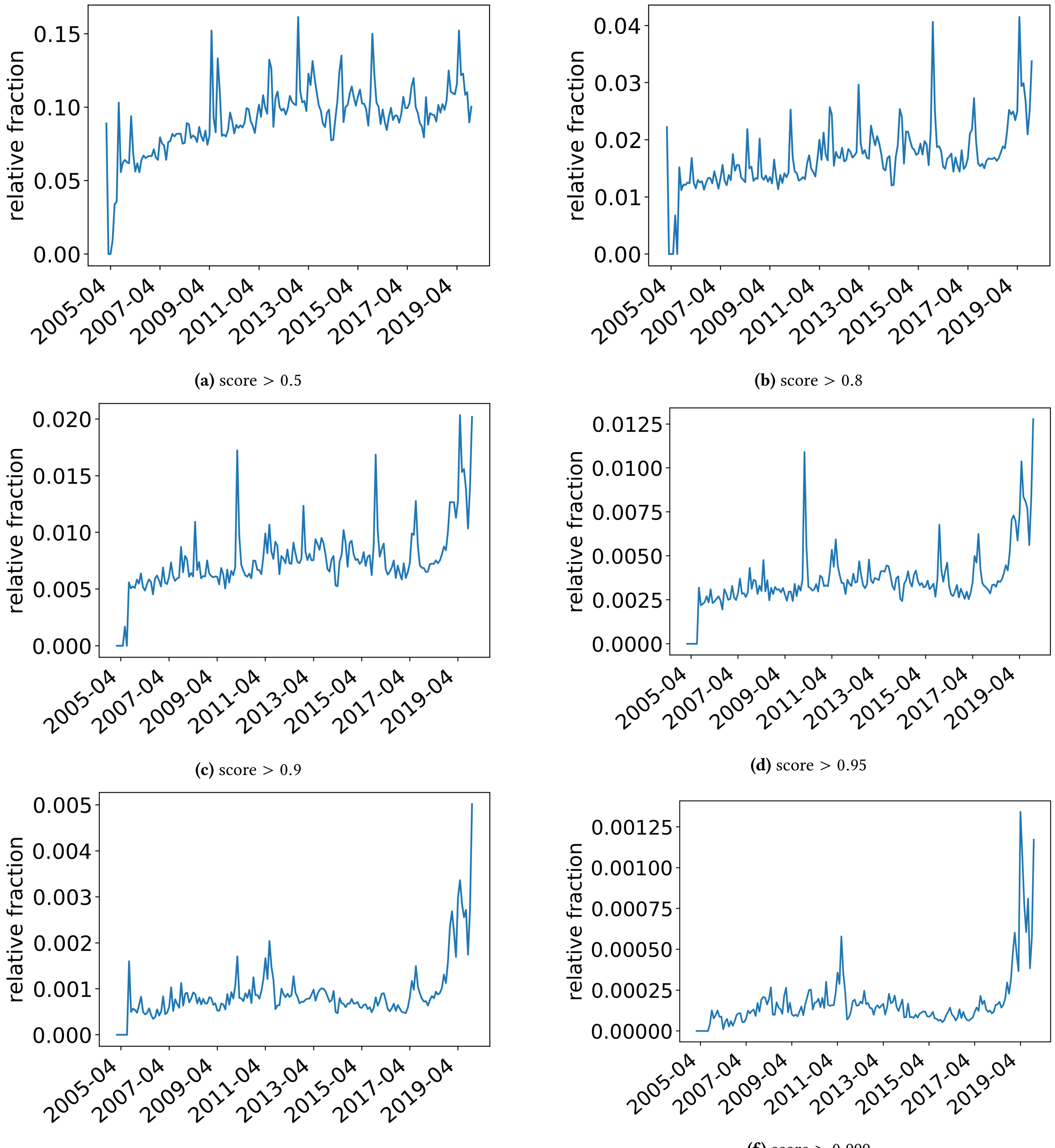
- **OpenAI's GPT-2 Detector.** We use OpenAI's publically available[3] detection model, a RoBERTa-large (356 million parameters) that was fine-tuned on GPT-2 generations using a mix of untruncated and top-$k$ 40 sampling. For each sequence no longer than 512 byte-level Byte-Pair-Encoding (BPE) tokens, the detector outputs the probability of that sequence being machine written. For longer texts, we use the score from the first 512-length sequence.
- **GLTR.** We follow the method proposed in Gehrmann et al. [13] but use the much larger GPT-2 XL model (1.5 billion parameters). For each token $T$ in the target text, we obtain a softmax distribution by conditioning the model on text surrounding $T$. Next, we compute the integer rank of $T$ in the sorted distribution. We then construct a document-level histogram over the individual token ranks binned into top-$k$ buckets, where $k \in \{10, 100, 1000, |V|\}$ and $|V| = 50257$ is the size of the token vocabulary. We train a logistic regression on this single 4-dimensional feature using the train split of GPT2-Output. Concretely, we take 250k human-authored documents and 250k GPT-2-generated documents split uniformly across the 8 settings.
- **Spam Baseline.** We train a spam / not-spam classifier using the Enron Spam Email dataset [22][4]. We construct train and test splits; train comprises 12875 documents for each spam and not-spam (a.k.a. ham) class, while test similarly comprises 3219 documents. Using the train split only, we learn a TF-IDF histogram featurizer using a vocabulary of 5000 lowercase words. We then train a logistic regression classifier on top of the 5000-dimensional featurized training documents. The featurizer and classifier combination achieves 96.9% accuracy on the test split. In the language quality evaluation described in the next section, we use the

[1] `https://commoncrawl.org/`.

[2] `https://github.com/rowanz/grover`

[3] `https://github.com/openai/gpt-2-output-dataset`

[4] `http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html`



**(a)** score > 0.5

**(b)** score > 0.8

**(c)** score > 0.9

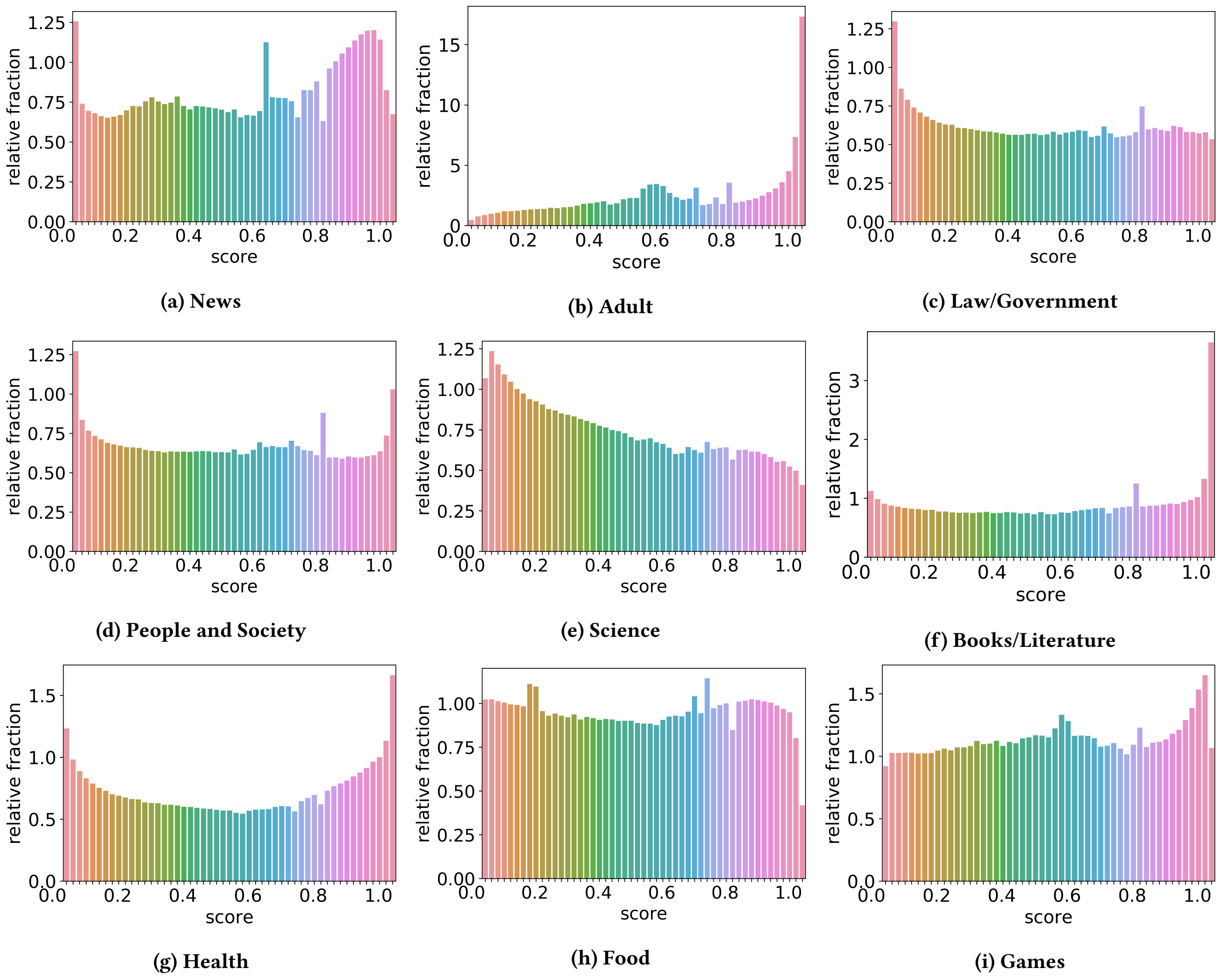
**(d)** score > 0.95

**(e)** score > 0.99

**(f)** score > 0.999

**Figure 3: Relative fraction of documents over time at various OpenAI detector score thresholds. For each threshold, the value at a particular date is the probability of observing a document scoring over the threshold on that date, conditioned on the date. We observe a burst in the fraction of low quality documents at the beginning of 2019.**

model’s estimate of the probability of not-spam as its effective language score. In other words, if the model estimates that a document is not-spam with probability 0.2, then its language quality score on this document is also 0.2.



(a) News

(b) Adult

(c) Law/Government

(d) People and Society

(e) Science

(f) Books/Literature

(g) Health

(h) Food

(i) Games

**Figure 4: Relative fraction of OpenAI detector scores for different topics. It is computed by dividing the score distribution conditioned on each category by the marginal score distribution over all categories.**

| Method | Corr. | 95% CI |
|---|---|---|
| OpenAI | **0.740** | **[0.637, 0.822]** |
| Spam Baseline | 0.454 | [0.309, 0.582] |
| GLTR LR | **0.713** | **[0.569, 0.835]** |
| Spam Baseline | 0.495 | [0.316, 0.659] |

**Table 1: Pearson correlation between human and classifier LQ scores. We observe a large, statistically significant, positive correlation for both models, indicating that they are effective predictors of language quality. Furthermore, in both cases, the correlation is stronger than the baseline, which was trained with supervision for spam detection.**

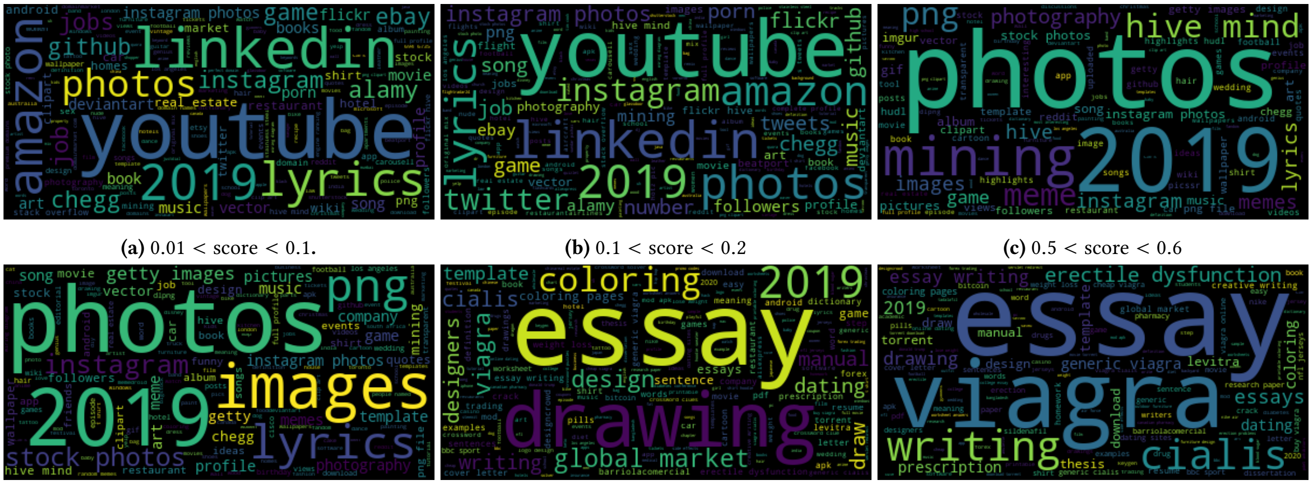
| | Cohen's kappa | 95% CI |
|---|---|---|
| GLTR LR | 0.501 | [0.387, 0.611] |
| OpenAI | **0.604** | **[0.474, 0.724]** |

**Table 2: Cohen's kappa coefficient for inter-rater reliability. The two raters achieve high, statistically significant agreement on the four possible LQ categories (including "Undefined").**

## 3.3 Language Quality Evaluation

As mentioned earlier, while we use "page" and "language" quality interchangeably, we are specifically interested in the quality of the language on the page. To that end, we define a language quality (LQ) score using the following criteria:

- 0: Low LQ. Text is incomprehensible or logically inconsistent.
- 1: Medium LQ. Text is comprehensible but poorly written (frequent grammatical / syntactical errors).



**(a)** 0.01 < score < 0.1.

**(b)** 0.1 < score < 0.2

**(c)** 0.5 < score < 0.6

**(d)** 0.6 < score < 0.7

**(e)** 0.99 < score < 1.0

**(f)** 0.999 < score < 1.0

**Figure 5: Word Cloud on Web500M for different ranges of OpenAI detector scores.**

- 2: High LQ. Text is comprehensible and reasonably well-written (infrequent grammatical / syntactical errors).
- Undefined: LQ is hard to assess for any reason.

We evaluate our machine vs. human classifiers and baseline using this criteria. To better assess language quality, we first filter Web500M by dropping all samples with fewer than 7.5k characters. Next, we define 3 buckets on the filtered Web500M corpus using percentiles of the classifier's P(machine-written) score: bottom = [0, 0.5], middle = [50, 50.5], top = [99.5, 100]. We sample 35 documents from each bucket for a total of 105 documents. Documents from the bottom, middle, and top buckets are assigned LQ scores of 2, 1, and 0 respectively. All documents are then rated by two human raters using the aforementioned criteria. Documents that at least one rater marked "Undefined" are dropped and all other documents are assigned a composite score that is the average of the raters' scores. Finally, we compute the Pearson correlation coefficient between the human and classifier's LQ scores along with a 95% bootstrap confidence interval. To measure the inter-rater reliability, or degree of agreement between the two raters, we compute Cohen's kappa coefficient along with a 95% bootstrap confidence interval.

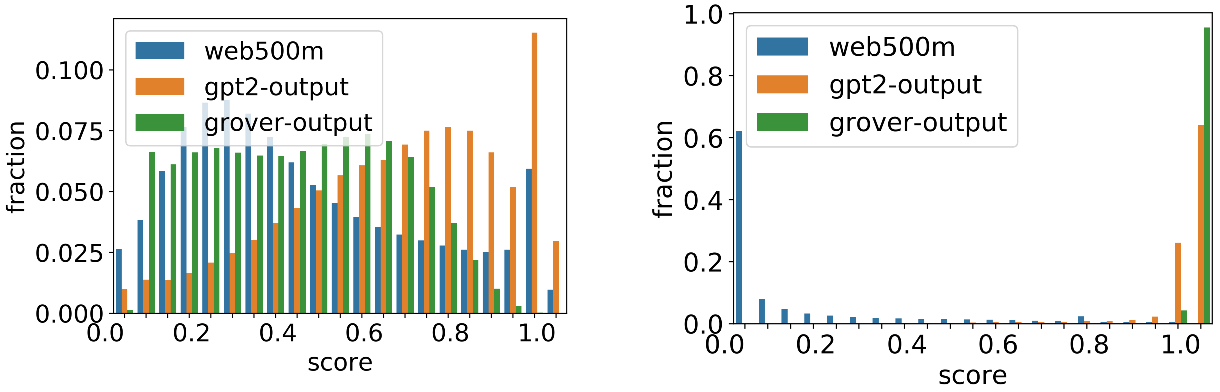
Correlation and inter-rater reliability results are shown in Table 1 and Table 2 respectively. Samples illustrating various language quality scores are shown in Figure 1. For both models, both the Pearson correlation and the inter-rater agreement is large and statistically significant, suggesting that documents with high P(machine-written) score tend to have low language quality. Furthermore, the models outperform the baseline, which was trained for spam detection in a supervised fashion. Machine authorship detection can thus be a powerful proxy for quality assessment. It requires no labeled examples - only a corpus of text to train on in a self-discriminating fashion. This is particularly valuable in applications where labeled data is scarce or where the distribution is too complex to sample well. For example, it is challenging to curate a labeled dataset representative of *all* forms of low quality web content.

| | GPT-2 | Grover |
|---|---|---|
| GLTR LR | 75% | 80% |
| OpenAI | **85%** | **82%** |

**Table 3: Test accuracy of models on different held-out distributions, each consisting of a balanced human / machine split. We observe, somewhat surprisingly, that the OpenAI detector generalizes to Grover generations better than the GLTR detector.**

## 3.4 Detector Performance

Table 3 shows test accuracy of the detectors on both GPT-2 and Grover distributions. Contrary to our intuition, we find that the OpenAI detector generalizes to the Grover distribution better than the simpler GLTR detector. In Figure 6 we compare the distribution of the detector scores on the web corpus against that of machine-written texts. Unlike the GLTR logistic regression detector, the OpenAI detector's score distribution is well-separated: scores are either either small or large. We focus on the OpenAI detector in subsequent analysis in light of its better predictive performance on machine vs. human discrimination as well as a higher correlation with human-rated LQ labels, as described in the previous section.



**(a) GLTR Logistic Regression**

**(b) OpenAI Detector**

**Figure 6: Score distributions for GLTR and OpenAI detectors respectively. The OpenAI detector separates web and machine-generated documents better than GLTR.**

## 3.5 Temporal Analysis

Using the OpenAI detector scores as a measure of language / page quality, we now characterize the temporal pattern of low quality content on the web. Is there more or less low quality content on the web recently? To control for the fact that the web corpus contains more documents published recently, we plot the ratio between the number of documents detected as low quality and the total number of documents for each historical month.

As visible in Figure 3, we observe a burst in the fraction of low quality documents at the beginning of 2019. One possible explanation for this is in the maturity of technology for fast and realistic-looking generation of text. We will later qualitatively describe these low quality pages.

## 3.6 Content Length Analysis

Figure 7 shows the fraction of documents with OpenAI detector scores in different ranges as a function of *untruncated* document length. Controlling for the fact that shorter documents are more common across the web, we find that low quality content tends to be shorter in length, peaking at about 3000 characters.

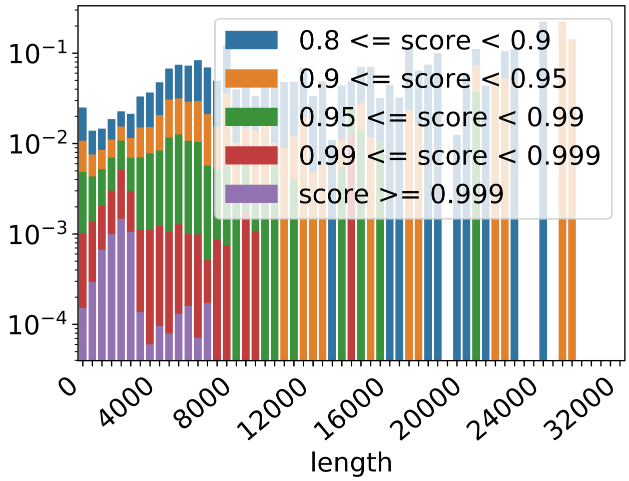


**Figure 7: Fraction of documents with OpenAI detector score in different ranges as a function of untruncated character length. Low quality content tends to be shorter in length, peaking at 3000 characters.**

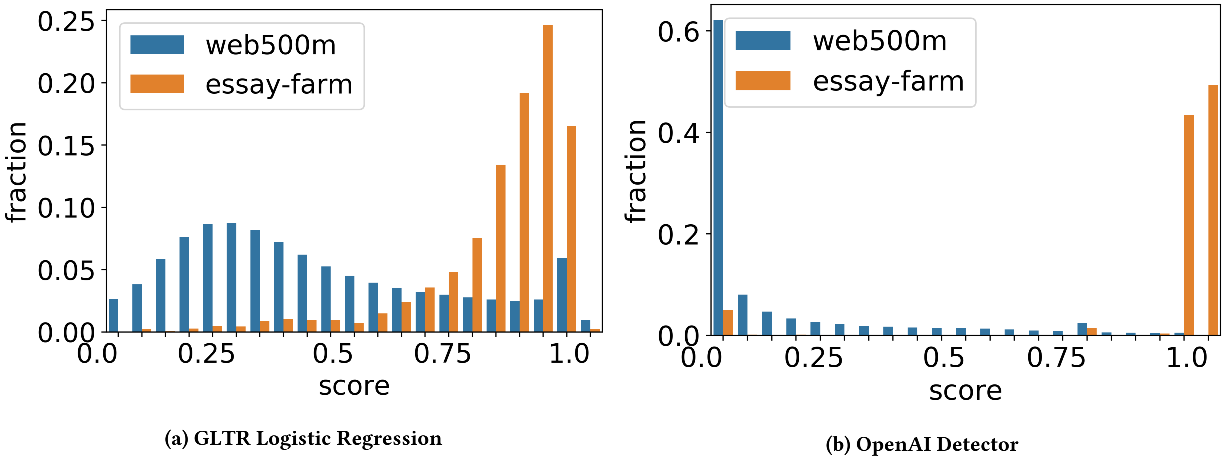
## 3.7 Topical Analysis

Figure 4 presents the distribution of OpenAI detector scores by document topic. Topics here are based on the "Content Categories" from the Google Cloud Natural Language API [5]. We analyze six high-level topic categories: News, Adult, Law / Government, People / Society, Science and Books / Literature. Based on our empirical results, we find that the score distributions vary significantly based on topic.

Among all topical distributions, we find that a large fraction of documents from the Adult category are flagged as low quality by our detector. Domains such as Games and Books / Literature also tend to be heavier on the low quality side of the histogram. Law / Government and Science topics trend lower with lower quality, suggesting these domains attract higher quality content creators. Food, interesting, is mostly uniform in nature while Health and People / Society follow a convex shape, with large numbers of documents clearly falling near 0 and 1.

We found it curious that the Books / Literature domain was flagged as low quality since we would expect it to consist

[5] https://cloud.google.com/natural-language/docs/categories



**(a) GLTR Logistic Regression**

**(b) OpenAI Detector**

**Figure 8: OpenAI detector score distribution on essay writing service domains. The domains have a high incidence of low quality content compared to the rest of the web.**

of high quality prose. Upon closer inspection, we found a slew of "essay farms", websites trying to sell possibly auto-generated essays to students. Moreover, the unusual number of documents near 1 for Health may be likely due to websites selling "adult health products".

### 3.8 Frequent Terms Analysis

In this section, we perform an analysis of the frequent terms from documents with different detector scores. In particular, we extract frequent terms for each document in Web500M and present word cloud visualizations for different score ranges.

Figure 5 illustrates the most frequent terms for six ranges of detector score: $[0.01, 0.1]$, $[0.1, 0.2]$, $[0.5, 0.6]$, $[0.6, 0.7]$, $[0.99, 1.0]$, and $[0.999, 1.0]$. Based on our observations, the topicality shifts drastically across score ranges. In the low-score range, the top frequent terms are ordinary web applications. However, as the scores approach 1, we notice heavy occurrences of NSFW terms. To ensure the word clouds do not contain inappropriate language, all clouds for score $> 0.5$ are computed by first filtering out documents that are marked as NSFW. Figures 5f and 5e are fairly representative of the documents found in these high-score ranges. We make several observations. Firstly, the emergence of the *"essay"*, *"writing"* and *"thesis"* keywords aligns well with the emergence of low quality documents in the Books / Literature topical distribution and is an indicator of the presence of essay farms. Secondly, we find keywords such as *"viagra"* which may explain the low quality peak we observe in the Health topical distribution.

### 3.9 Qualitative Analysis

This section presents key qualitative insights into the kind of web documents our model deems low quality. We manually inspected the top-0.01% scoring documents.

- **Machine Translated Text** - We found web documents that look like they might have been translated from English to another language and then back to English.
- **Essay Farms** - We found essay farms selling writing services. Figure 8 shows the score distribution of pages on a set of essay writing service domains. It's conceivable that some of these pages were machine-generated, although not necessarily by neural generative models.
- **Attempts at Search Engine Optimization (SEO)** - Documents that attempt to perform SEO tend to be flagged as very low quality. This is intuitive since these texts tend to simply string a series of keywords together and are therefore incoherent. Furthermore, we found a moderate number of product pages and professional profiles that also attempt to perform some form of SEO. We observed that media-centric domains, such as image hosting domains, often contain incomprehensible embedded text, possibly for SEO.
- **NSFW (Not-Safe-for-Work) Content** - We observed that a lot of low quality pages contained copious amounts of NSFW content. Upon deeper inspection, we found that many NSFW pages embed long paragraphs of nonsensical text as hidden text. The textual content is generally NSFW and incoherent. We speculate that this might also be an attempt at SEO.

## 4 CONCLUSION

This paper posits that detectors trained to discriminate human vs. machine-written text are effective predictors of webpages' language quality, outperforming a baseline supervised spam classifier. We substantiate this through rigorous human evaluation and then apply these low language quality detectors on half a billion webpages. We observed interesting topical and temporal patterns to the low quality content and discovered that many offenders are either (1) machine-translated text, (2) essay farms, (3) attempts at search engine optimization, or (4) NSFW content. We hope researchers interested in text quality find our web-scale analysis useful. Furthermore, we hope they leverage the insight that a reasonable language-quality classifier can be constructed with nothing more than a corpus of human text: train a generative model on the corpus, use it to synthesize machine text, and finally train a model to discriminate between the natural text and synthetic machine text.

## REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.

[2] Danial Alihosseini, Ehsan Montahaei, and Mahdieh Soleymani Baghshah. 2019. Jointly Measuring Diversity and Quality in Text Generation Models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*. Association for Computational Linguistics, Minneapolis, Minnesota, 90–98. https://doi.org/10.18653/v1/W19-2311

[3] Dimitrios Alikaniotis and Vipul Raheja. 2019. The Unreasonable Effectiveness of Transformer Language Models in Grammatical Error Correction. *arXiv preprint arXiv:1906.01733* (2019).

[4] Ioannis Arapakis, Filipa Peleja, Barla Berkant, and Joao Magalhaes. 2016. Linguistic Benchmarks of Online News Article Quality. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1893–1902. https://doi.org/10.18653/v1/P16-1178

[5] 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*.

[6] Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or Fake? Learning to Discriminate Machine from Human Generated Text. *arXiv preprint arXiv:1906.03351* (2019).

[7] Michael Bendersky, W Bruce Croft, and Yanlei Diao. 2011. Quality-biased ranking of web documents. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 95–104.

[8] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).

[9] Gordon V Cormack, Mark D Smucker, and Charles LA Clarke. 2011. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval* 14, 5 (2011), 441–465.

[10] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and Play Language Models: a Simple Approach to Controlled Text Generation. arXiv:1912.02164 [cs.CL]

[11] 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* (2018).

[12] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833* (2018).

[13] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. GLTR: Statistical Detection and Visualization of Generated Text. *arXiv preprint arXiv:1906.04043* (2019).

[14] EA Grechnikov, GG Gusev, AA Kustarev, and AM Raigorodsky. 2009. Detection of artificial texts. *RCDL'2009 Proceedings. Petrozavodsk* (2009), 306–308.

[15] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*. 1693–1701.

[16] Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751* (2019).

[17] Huggingface. 2019. Write with transformer. 2019. (2019). https://transformer.huggingface.co/

[18] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* (2019).

[19] Wouter Kool, Herke Van Hoof, and Max Welling. 2019. Stochastic Beams and Where to Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. *arXiv preprint arXiv:1903.06059* (2019).

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[21] Mohsen Mesgar and Michael Strube. 2018. A neural local coherence model for text quality assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4328–4339.

[22] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. 2006. Spam filtering with naive bayes-which naive bayes?. In *CEAS*, Vol. 17. Mountain View, CA, 28–69.

[23] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. 2006. Detecting spam web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web*. 83–92.

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 311–318.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[26] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. (2019).

[28] John Seabrook. 2019. The Next Word. *The New Yorker* (2019), 52–63.

[29] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203* (2019).

[30] TabNine. 2019. Autocompletion with deep learning. 2019. (2019). https://tabnine.com/blog/deep/

[31] Yi Tay, Dara Bahri, Che Zheng, Clifford Brunk, Donald Metzler, and Andrew Tomkins. 2020. Reverse Engineering Configurations of Neural Text Generation Models. *arXiv preprint arXiv:2004.06201* (2020).

[32] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424* (2016).

[33] Nick Walton. 2019. AI Dungeon. 2019. (2019). http://www.aidungeon.io/

[34] Nick Walton. 2019. GPT-2 Neural Network Poetry. 2019. (2019). https://www.gwern.net/GPT-2

[35] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319* (2019).

[36] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* abs/1910.03771 (2019).

[37] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).

[38] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending Against Neural Fake News. *arXiv preprint arXiv:1905.12616* (2019).

[39] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1097–1100.