New-Wiki Eval: An Evolving Wikipedia Multi-metric Evaluation for Large Language Models

Anonymous submission

Abstract

Recent progress in natural language generation enables models such as GPT-3 to generate long text that makes human annotations unreliable. Although the generated text from generative large language models is fluent, it is often not knowledgeable and informative enough. Existing automatic evaluation metrics like BLEU and ROUGE scores have their own shortcomings for evaluating open-ended generation. More importantly, no good evaluation datasets exist for an open-ended generation. In this paper, we propose the task "Wikipedia generation" and a set of evaluation metrics to help researchers evaluate their model performance on knowledge-intensive long text generation from seven perspectives. We notice Wikipedia is good reference text since it is knowledgeable, proofread, and constantly evolving. Thus, we collect and release the New-Wiki dataset as our evaluation set and will keep it updated to provide an out-of-sample evaluation set. Then, we evaluate state-of-the-art language models including GPTs, BLOOM, OPT, BART, and T5 using our metrics. We then show the correlation of our proposed metrics with prior research and find insights into different types of generative language models.

Introduction

Recently, natural text generation (NLG) models have made significant breakthroughs driven by the encoder-decoder paradigm (Sutskever, Vinyals, and Le 2014), the powerful transformer architecture, and the growing computing resources (Vaswani et al. 2017). In particular, GPT-3, a large language model (LLM) consisting of 175 billion parameters, is now able to generate human-indistinguishable articles of any given prompt (Brown et al. 2020). Although various automatic evaluation metrics have been proposed for open-ended generation, they are not comprehensive and reliable enough given the creativity and complexity of the generated passage. Therefore, when it comes to evaluating long-text generation such as news article generation and story generation, the researchers often crowdsource human evaluation, asking the annotators to distinguish between human written text and generated text (e.g., GPT-3). As Sellam, Das, and Parikh (2020) pointed out, conducting crowdsourcing experiments is an expensive and high-latency process, which prohibits NLG researchers from getting immediate feedback for their generation models. Thus it is necessary to create well-rounded and robust automatic evaluation metrics.

In this paper, we propose a set of well-rounded automatic evaluation metrics for knowledge-intensive long text generation. As large language models are shown to be able to generate fluent and reasonable open-ended news articles and stories (Zellers et al. 2019), researchers become interested in how to make the generated text factually correct. Knowledge-intensive text generation aims at generating informative, logical, and factual passages that could show the model's understanding of commonsense and real-world knowledge. Previous work has shown that large language models can store real-world knowledge into their parameters (Roberts, Raffel, and Shazeer 2020), researchers are now trying to enable the model access, precisely manipulate knowledge, and then generate text based on a knowledge base (Komatsuzaki 2020; Guu et al. 2020; Lewis et al. 2020). However, most of knowledge-related evaluation tasks are still benchmark based and not intrinsic. This means it is likely to suffer from the issue of data leakage. To solve this issue, we suggest taking advantage of the fact that Wikipedia is added constantly and could serve as an evolving test set.

In this paper, we propose an evaluation framework based on Wikipedia to help researchers assess the model's performance on knowledge-intensive article generation. We first note that Wikipedia is naturally a great reference text for this task since it is knowledgeable, well-written, and most importantly, constantly evolving. The trait of constantly evolving is a key to our research since it provides a way to separate out a test set that the newest large language models are not trained on. Thus, we collect and open-source New-Wiki Dataset consisting of Wikipedia articles created after June 2021.

We then propose a new task called Wikipedia generation, in which we let the language models generate Wikipedia-style articles. We expect a good language model to generate text, similar to a real Wikipedia article, to be knowledge-intensive, factually correct, and relevant to. Thus we purpose a suite of evaluation metrics from six aspects to measure the generated articles. Lastly, we conduct extensive experiments by using 7 state-of-the-art large language models to do the Wikipedia generation task. We show that our evaluation metrics provide a well-rounded evaluation that is highly correlated with human evaluation. Then we also find new insights into the characteristics of different state-of-the-art language models and how far away the language models are in the

task of generating knowledge-intensive long articles.

To summarize, our contributions are: 1) Open-sourced New-Wiki Dataset for knowledge-intensive long text generation 2) Observe the "evolving" trait of Wikipedia and purpose the evaluation framework based on that 3) Purpose a set of well-rounded automatic evaluation metrics for Wiki-style text generation 4) Conduct extensive experiments to show the correlation with human judgments and make experimental observations into state-of-the-art generative language models

Related Work

Dataset for Evaluating LLMs

Evaluating multi-functional large language model is a long-existing problem and nowadays, researchers have collected various evaluation datasets for most of the downstream tasks. For example, LLMs are examined of its Question and Answering ability on WebQuestions (Berant et al. 2013) and Natural Questions (Kwiatkowski et al. 2019), its reading comprehension skill on SQuAD (Rajpurkar, Jia, and Liang 2018), and translation quality on WMT dataset (Barrault et al. 2019). However, there is no dataset proposed for evaluating LLMs' open-ended generation skills because of the creativity of the generated text and the hardness of finding out-of-sample reference text. Our New-Wiki dataset would provide an evolving evaluation dataset for the open-ended generation.

Automatic Evaluation of Open-ended Text Generation Many evaluation metrics for evaluating long-text generation have been proposed. They could be categorized into n-gram based unconstrained metrics and deep learning-based metrics. N-gram based metrics including BLEU score (Papineni et al. 2002), ROUGE score (Lin 2004) are sensitive to lexical differences but could not capture semantic variations from the reference text. However, in the task of Wikipedia generation, having lexicon overlap is very difficult and thus n-gram based evaluation is hard as we show later in the paper.

Various deep learning-based metrics for NLG have been proposed recently. For example, BERTScore measures the similarity between the generated text and the reference text (Zhang et al. 2019). BARTScore treat the evaluation task as a generation task and obtained more robust metrics (Yuan, Neubig, and Liu 2021). Although these metrics have been shown to be able to provide a single score for the generated text, they are black-box models that cannot further explain how different aspects of the generation contribute to the scoring. This hinders researchers from getting a deeper understanding of the LLMs. In our research, we carefully choose our automatic evaluation metrics from six different aspects to construct a well-rounded and more explainable evaluation framework.

Human Evaluation of Open-ended Text Generation While most researchers are justifying their models by collecting crowdsourced human evaluations, it's an expansive and low-latency evaluation method (Sellam, Das, and Parikh 2020). Even worse, given the length and complexity of generated text, many Amazon Mechanical Turk workers cannot fully read the text they are reading and even expert raters struggle to read and rate model-generated text (Karpinska, Akoury, and Iyyer 2021). For example, In GPT-3's crowd evaluation, the annotator's accuracy in identifying whether short news articles are model generated or human written drops to 52%, barely better than random guessing (Brown et al. 2020).

The scene behind using crowdsourced human evaluation is the fact that automatic evaluation of long generated text is challenging given it is completely open-ended. Researchers purpose to evaluate long text generation via lexical cohesion (Lapata, Barzilay et al. 2005), writing style consistency (Roemmele, Gordon, and Swanson 2017), lexical overlap with multiple references (Holtzman et al. 2019). Compared to prior works, we consider multiple aspects of the generated text and provide a systematical evaluation. Furthermore, the task of Wikipedia article generation provides an ideal setting for evaluating the knowledge or factuality of the generated text, which is not studied in prior work.

Knowledge in Language Generation While GPT models are capable of generating long and complex open-ended articles, they are mostly factually incorrect. Researchers are trying to incorporate real-world knowledge and commonsense into the language models. Gardent et al. (2017) proposed the KB-to-text generation problem, aiming at converting a discrete knowledge base into long text. Another interesting line of research tries to let the model search over the database and extract the answer through information extraction. Building on Retrieval-augmented language model pretraining (REALM; Guu et al. 2020), Retrieval-augmented Generation (RAG; Lewis et al. 2020) is able to first retrieve relevant passages from knowledge base and generate answer from the selected passages. However, it is not able to generate long articles as of now and thus not included in this paper.

New-Wiki Datset

Finding a good evaluation dataset for open-ended generation is a challenging problem given the creativity of the generation and the hardness of finding good reference text. We choose Wikipedia for the following considerations: 1) Wikipedia contains a set of factual knowledge that can be considered as "ground truth". This provides the information that is expected to show up in the generation and thus better to serve as a benchmark compared to intrinsically open-ended task like story generation. 2) Wikipedia and constantly evolving. This helps to avoid the issue of data leakage that might boost the model performance significantly (Elangovan, He, and Verspoor 2021).

Thus we collect and open-source a Newly Created Wikipedia Dataset (New-Wiki) consisting of Wikipedia articles created after June 2021 as our test set.[1] While language models keep evolving and will be trained on newer Wikipedia, we keep New-Wiki updated regularly and thus it could serve as a good test set of knowledge-intensive text

[1]We choose all articles after June 2021 to make sure GPT-3 Davinci 002 is not trained on them.

generation without the issue of data leakage. We also note that although the language model is not supposed to know about future and have direct knowledge of the Wikipedia article (which is created after the model's release). Since we include the title as well as the first sentence of the article, the model should be able to get enough context to infer about the content of the Wikipedia article.

In practice, we used Wikipedia API and requested all the Wikipedia articles that are created between June 2021 and December 2021. We filtered out articles that have less than 10 revisions to make sure the article is a refined article. We then filtered out articles that is shorter than 500 words to ensure these articles are good reference text for long text generation. We sample 3000 articles from these articles. Finally, we split the raw articles into prompt and completion by concatenating the title and the first sentence of the article as the prompt and the rest of the article as the completion. The final dataset that is summarized in Table 1.

Entity Type	Occurrence	Percentage
Human	1328	44.2%
Taxon	251	8.4%
Media	239	8.0%
Event	217	7.2%
Human Seattlement	185	6.2%

Table 1: Topics covered in New-Wiki

Methodology

Generative Language Models

In this paper, we evaluate the following state-of-the-art generative language models: GPT2 (Radford et al. 2019), GPT3 (Brown et al. 2020), OPT (Zhang et al. 2022), BART (Lewis et al. 2019), T5 (Raffel et al. 2019), GLM (Du et al. 2021), and BLOOM (BigScience 2022). For BART and T5, we fine-tune them on a about 2000 Wikipedia articles for 10 epochs to let them to perform long text generation. For GPT3, we fine-tune it using OpenAI's fine-tuning API. Models are summarized in Table 2.

Model	# Parameters	Release date
GPT2	1.5B	Feb. 2019
BART-base	110M	Oct. 2019
T5-base	220M	July 2020
GPT3	175B	July 2020
OPT-66B	66B	May 2022
BLOOM	175B	June 2022
GLM	130B	Aug 2022

Table 2: The release date and parameters of SOTA large language models

Evaluation Metrics

We select evaluate metrics from six different perspectives so that we get a well-rounded and explainable view of the language model's performance.

Text Complexity Text complexity is an important aspect of evaluating generated Wikipedia articles. One would expect a good Wikipedia article to have reasonably high text complexity. Average sentence length and Frequency of complex word usage are intuitive measures of the text complexity. Building on these metrics, In Kincaid et al. (1975), Flesch–Kincaid readability score (FK-score) computes a weighted and normalized score to indicate how difficult an English passage is to understand. The formula is given by

$$\text{FK-score} = 206.8 - 1.015 * \frac{|\text{words}|}{|\text{sentences}|} - 84.6 * \frac{|\text{syllables}|}{|\text{words}|} \quad (1)$$

Similarly, Gunning fog index (Wikipedia 2022) estimates the years of formal education a person needs to understand the text on the first reading by the following formula.

$$\text{Gunning fog index} = 0.4 * \frac{|\text{words}|}{|\text{sentences}|} + 100 * \frac{|\text{complex words}|}{|\text{words}|} \quad (2)$$

Text Quality The quality of generated text is also an important aspect for evaluating generative language models. We use a trained LSTM model to score the generated text. The LSTM model is trained on more than 10,000 student essays and its human-graded score which to assess the quality of the essay in response to the prompt. (Khushali Thakkar 2019)

Specificity Specificity quantifies the level of detail in the text and the organization of the information. For example, in Ko, Durrett, and Li (2019), they provide a good example to help readers understand specificity intuitively, where Example 2 clearly contains more detail of the subject comparing to Example 1.

> Ex1: This brand is very popular and many people use its products regularly.
> Ex2: Mascara is the most commonly worn cosmetic, and women will spend an average of $4,000 on it in their lifetimes.

Such a metric helps us to see whether the language model is getting into the details of the subject or just piling up unrelated terms to make the text seemingly compiling.

We adopt the LSTM-based model from Ko, Durrett, and Li (2019) to estimate the specificity of the generated text. In our task, one would expect a good generated Wikipedia article to be specific and mention more details of the subject.

Diversity A good Wikipedia article should contain diverse lexicon to describe the subject. To measure the lexical diversity, we introduce the distinct-n metric introduced in (Li et al. 2015). Distinct-n counts the number of distinct word or 2-grams in the passage and thus captures the diversity of

words or 2-grams. It's given by the following formula where $|\cdot|$ denotes the cardinality or count and n equals 2.

$$\text{Distinct-n} = \frac{|\text{unique n-grams}|}{|\text{words}|} \tag{3}$$

Repetition Although the noxious problem of repetition is getting less prevalent as the model size grows, given the difficulty of Wikipedia generation task, from time to time, there are still repetitions in GPT2 and GPT3. Thus we include the repetition metric to assure the generated text is not repeating itself. Thus we use the rep-n score from (Welleck et al. 2019) to measure the number of repeated n-grams in the generated text. In our experiment, we take n equals 4 to captures the repetition of longer text. The formula is given by

$$\text{rep-n} = 1.0 - \frac{|\text{unique n-grams}|}{|\text{n-grams}|} \tag{4}$$

Information Density Given our task of generating a knowledge-intensive articles like Wikipedia, evaluating whether the model could generate informative text is important. To measure the informativeness, we purpose the information density metrics. We use spacy to do Named Entity Recognition to extract the entities and then calculate it by the following formula.

$$\text{Information Density} = \frac{|\text{entities}|}{|\text{words}|} \tag{5}$$

Relevance The relevance between the Wikipedia articles and generated text is a crucial component of our evaluation metrics. We use S-BERT score and entity overlap to calculate their relevance. We first purpose Entity overlap metric which intuitively gives a score between 0 and 1 that measures the number of entities mentioned in the generated text and the reference text. It is calculated by the following formula.

$$\text{Entity Overlap} = \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|} \tag{6}$$

E_1 represents the entities in the generated text and E_2 represents the entities in the Wikipedia article. We believe entities including certain terminology, people's name, locations, etc. are good indications of the knowledge. Thus we use entity overlap to measure the knowledge of the model.

However, we note that the entity metrics require the and thus synonyms or different form of the word would be overlooked. Thus we use the S-BERT score (Reimers and Gurevych 2019) to capture the semantic similarity between the generated text and the original Wikipedia. It is calculated by first using Sentence-BERT to embed the articles into fixed-length vectors and then compute the inner product.

We are not using traditional measures of relevance like BLEU or ROUGE because getting n-gram overlap between open-ended generation is very difficult and results in BLEU score near 0. BLEU score calculated using ScareBLEU is reported in the Appendix (under the scale of 100).

Experiments

For long-text generation, we select the 2000 longest Wikipedia articles from New-Wiki as our evaluation set. We generated Wikipedia using the generative language models discussed above to perform text completion. Specifically, we let each model to generate 20 completions for one prompt and then we select the top 10 generated text by its word count to filter out empty and short completions. We also store the original Wikipedia text as the reference text ground-truth for comparison with generated text. Finally we apply our evaluation metrics to study the performance of generative models.

We conduct the following three experiments: 1) We experiment with different language models including GPT-3, fine-tuned GPT-3, GPT-2, BART, T5, OPT, GLM and BLOOM to show the characteristic of these models. We fix the model hyper-parameters to top_k = 20, top_p = 0.9, temperature = 0.9. 2) We conduct an ablation study of the GPT-2 models under different parameter settings. We did a grid experiment by choosing top_p = [0.5, 0.9, 0.95, 1.0], top_k = [20, 50, 100, 500], and temperature = [0.1, 0.5, 0.9] 3) We conduct a comparison of model performance on old vs new Wikipedia articles. For old Wikipedia generation, we randomly select 2000 articles from older Wikipedia that are longer than 400 words and went through the same generation process. We fix the parameter setting with top_p = 0.9, top_k =50, and temperature = 0.9.

Results

With 7 generative models and 7 evaluation metrics, we conduct a thorough evaluation current state-of-the-art language models. Full results are available in Appendix.

To provide better visualization of the experiment results, histograms in Figure 1,2,3,5 are rescaled into 0 and 1. For mean value, we handcraft the range of the metrics and then use the min-max scaler to rescale them. We also draw the range of plus or minus one standard deviation. We set FK-Score $\in [30, 60]$, essay score $\in [4.5, 5.5]$, relevance $\in [0, 1]$ S-BERT $\in [0, 1]$, information density $\in [0, 0.5]$, gunning-fog $\in [10, 25]$, distinct-n $\in [0.5, 1]$. For the standard deviation of the metrics, we directly use rescaled it to 0 and 1 using the min-max scaler.

Correlation with prior research

We first experiment with the different decoding mechanisms and parameters to show that our evaluation metrics would provide results that highly correlate with prior research. This validates the effectiveness of our evaluation framework.

Nucleus sampling We found that when increasing the top-p value, distinct-n, essay score, and text complexity scores would grow while the relevance score and rep-p metric would decrease. This is consistent with the design of nucleus sampling where a high top-p value leads the model to output tokens with lower probability and often harder and unexpected.

We also note that lower top-p value leads to higher rep-n which then leads to more extreme text and thus lower top-p value corresponds to a higher overall standard deviation although the standard deviation for each prompt is lower by design. This phenomenon is also observed for top-k and temperature sampling.

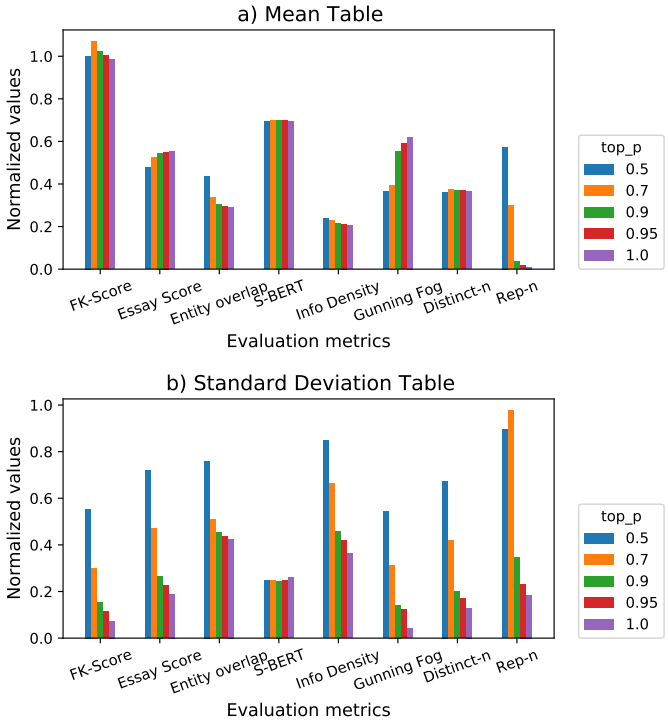

Figure 1: Mean and standard deviation of the evaluation metrics when changing the top-p value.

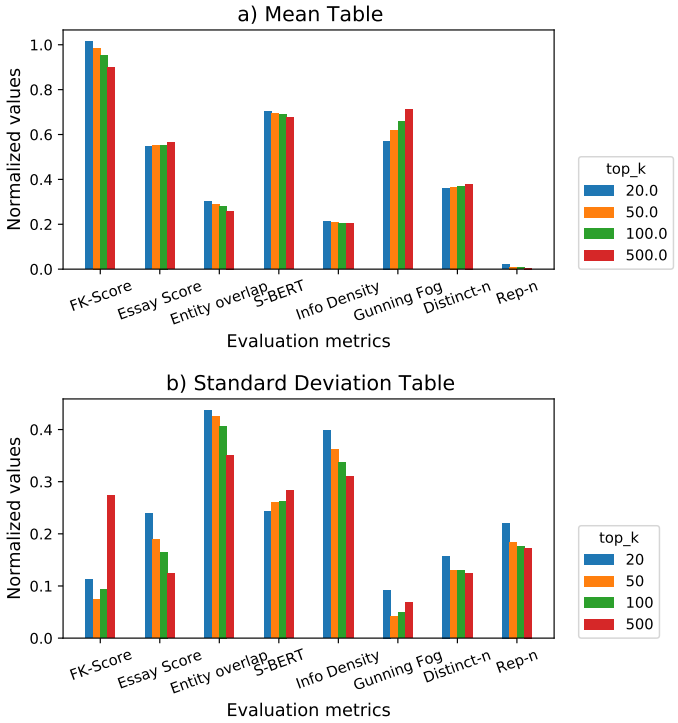

Figure 2: Mean and standard deviation of the evaluation metrics when changing the top-k value.

Top-k sampling When we change the hyperparameters of top-k value and temperature, the change in performance of the language model is small and less consistent. This leads us to believe that top-k sampling method has relatively small effect on the generated text and thus is not a less effective decoding mechanism comparing to nucleus sampling, similar to the argument made in (Holtzman et al. 2019).

Temperature Temperature appears to be the parameter that has the most significant effect on GPTs generation. When we increase temperature, the essay score and distinct-n metric increase by a large amount, while the relevance score decreases significantly compared to top-p and top-k sampling. This is consistent with the design of temperature where the model with high temperature is expected to be more creative and decodes tokens that are less expected tokens (and often less frequent and harder words). This is similar to the prior observation that when lowering temperature improves generation quality, it decreases the text diversity (Caccia et al. 2018).

Independence of Evaluation Metrics

In Figure 4, we present the correlation matrix across our metrics. We find that majority of the metrics in our evaluation framework are weakly correlated. This shows we successfully select evaluation metrics from different perspectives and each metric could tell a relatively independent characteristics of the LLM.

The only two set of metrics are highly correlated is text

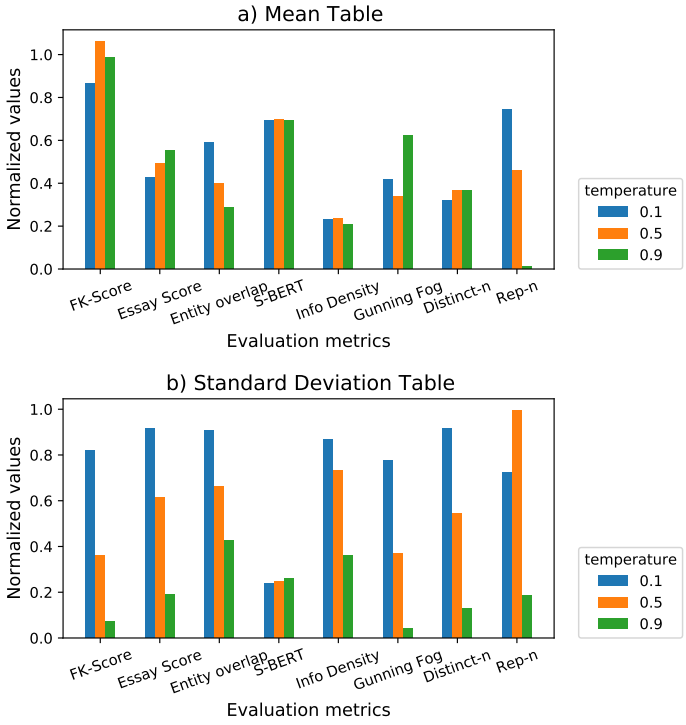

Figure 3: Mean and standard deviation of the evaluation metrics when changing the temperature

complexity (FK-Score and Gunning Fog Index) and text diversity (distinct-n) as both perspectives would favor harder words. The strong correlation between Flesch Score and Gunning Fox Index and the medium correlation between S-BERT and Entity Overlap that our choice to put them under the same metric. The high repetition score hurts the model performance as expected since it is negatively correlated with relevance, essay score, and information density.

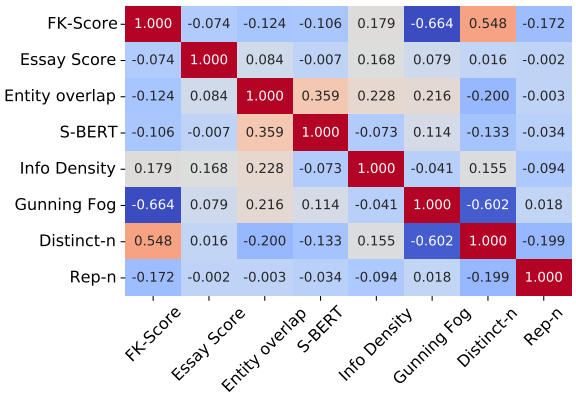

Figure 4: Correlation matrix across metrics

New insights of large language models

We further investigate the characteristics of different language models by letting all models listed to do Wikipedia generation under the same set of parameters. We then make the following observations.

GPT-3 GPT-3's generated text is better than all other models based on our metrics. Table 3 shows that GPT-3 has the highest Entity Overlap, S-BERT score, information density, and top 3 Essay Score and text complexity. In terms of the relevance metrics, GPT-3 achieves an extremely high entity overlap score of 0.778, meaning that the majority of the entities in real Wikipedia are also mentioned in GPT-3's generation. This shows us GPT-3's ability to first store Wikipedia-level knowledge into its parameter and then retrieve it during generation. However, we also note that although GPT-3 is getting good scores, it is actually getting more complex and harder to read than real Wikipedia articles measured by the gunning fog score, FK-score, and information density.

BART and T5 vs. GPT-2 BART and T5 model has very different characteristics from GPTs. BART generates significantly harder words (distinct-n ↑) and harder text (gunning fog ↑). Having higher essay score shows that these complicated words are composed together correctly but write hallucinating passages as BART has the lowest relevance score. On the other hand, T5 generates simpler text (low information density and essay score) but its relevance score is significantly lower than all GPTs. Table 2 shows that BART and T5 have smaller amount of parameters compaing to GPT-2 and thus we believe this is a good example of larger language models being able to store more world knowledge.

OPT, BLOOM, GLM vs. GPT-3 OPT, BLOOM, and GLM are state-of-the-art LLM released in 2022. Table 3 shows that their performance is indeed better than older version of LLMs in most of the dimensions. Among these four, one can see that GPT-3 Davinci is doing the best job, with notably higher score in entity overlap and S-BERT score. For OPT-66B and BLOOM, they are performing reasonably good with high text complexity and good relevance score between GPT-2 and GPT-3. So we conclude these are valuable open-source model that have open-ended generation ability in between GPT-2 and GPT-3. Also we note that the high rep-n score for GLM indicates it is making low-quality generations. And this is consistent with our manual checking where we found sentence repetitions and trailing symbols.

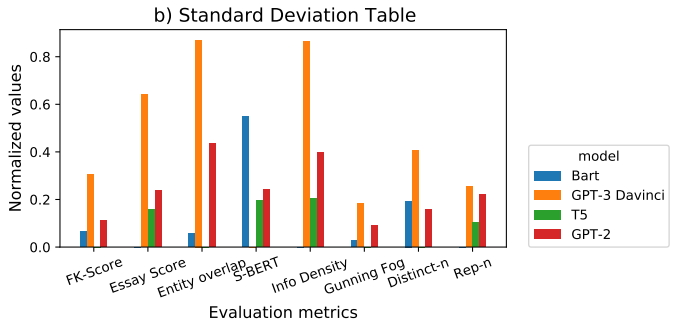

Figure 5: Standard deviation of evaluation matrix of different models

LLM's stability Although a larger model like GPT-3 has a higher score, it also has a higher score standard deviation, indicating larger models are actually less stable. Figure 5 shows standard deviation of metrics roughly follow this pattern: GPT-3 > GPT-2 > BART > T5. This is roughly the order of the number of parameters of these generative models. This observation leads us to hypothesize that since larger models like GPT-3 are more knowledgable, they would have enough knowledge and thus the confidence to "take the risk" and output something that is more specific and risky.

However, as shown in Table 3, the specificity score decreases as we use larger language models and disprove our hypothesis. We suspect it is because the deep learning-based specificity model fails to generalize to Wikipedia generation. And we need domain-specific labels to fine-tune the model to get a good evaluation. We believe specificity is an important metric when evaluating Wikipedia-style generation. And we will continue on finding good specificity estimation with better models or fine-tuning data.

Presence and Frequency Penalty When we increase the presence and frequency penalty, we penalize the model for generating tokens that have been used and thus force the model to change topics frequently. Thus, we see a significant drop in the relevance between the generation and real Wikipedia (entity overlap score ↓, S-BERT score ↓) and see an increase in the word diversity (distinct-n ↑). What's interesting is that applying the presence penalty and frequency penalty also hurts the quality of generated text as essay

Model	FK-Score (↓)	Essay scoring	Entity Overlap	S-BERT	Info density	Gunning_fog	Distinct_n	Rep_n	Specificity
Bart	**30.503**	5.159	0.216	0.57	0.131	**24.129**	0.692	0.004	0.543
T5	57.491	5.012	0.227	0.669	0.108	18.937	0.676	0.010	0.532
GPT-2	60.456	5.048	0.3	0.702	0.106	18.563	0.679	0.021	0.502
Fine-Tuned GPT-3 Curie	52.382	5.101	0.511	0.757	0.134	21.99	0.673	0.005	0.47
GPT-3 Curie	49.032	5.134	0.746	0.76	0.15	22.892	0.655	0.009	-
OPT-66B	53.741	5.114	0.324	0.72	0.118	19.416	0.702	0.035	-
GLM	50.812	**5.157**	0.291	0.692	0.122	19.882	0.543	0.208	-
BLOOM	55.794	5.081	0.249	0.603	0.112	19.523	**0.694**	0.036	-
GPT-3 With Penalty	50.161	5.119	0.242	0.628	0.142	22.689	0.661	0.006	-
GPT-3 Davinci	47.502	5.139	**0.778**	**0.762**	**0.153**	23.503	0.639	0.008	0.44
Wikipedia	52.646	5.057	1.000	1.000	0.111	21.424	0.692	0.007	-

Table 3: Mean of evaluation metrics of different LLM

Model	FK-Score (↓)	Essay scoring	Entity overlap	S-BERT	Info density	Gunning fog	Distinct_n	Rep_n	Top_p	Top_k	Temp	Wiki time
GPT-2	**60.711**	5.046	0.304	0.701	0.108	**18.295**	0.685	0.035	0.9	50	0.9	new
GPT-2	61.651	**5.061**	**0.354**	0.700	0.107	18.024	0.695	0.035	0.9	50	0.9	old
GPT-2	**60.127**	5.050	0.296	0.698	0.106	**18.858**	0.684	0.018	0.95	50	0.9	new
GPT-2	60.987	**5.062**	**0.35**	0.699	0.105	18.607	0.693	0.017	0.95	50	0.9	old
GPT-2	**59.574**	5.051	0.288	0.695	0.103	**19.311**	0.683	0.01	1	50	0.9	new
GPT-2	60.274	**5.063**	**0.339**	0.694	0.103	19.175	0.692	0.008	1	50	0.9	old
GPT-2	**58.568**	5.053	0.279	0.688	0.102	**19.875**	0.684	0.007	1	100	0.9	new
GPT-2	59.289	5.068	**0.328**	0.688	0.102	19.651	0.693	0.005	1	100	0.9	old
GPT-2	**56.956**	5.065	0.259	0.675	0.102	**20.685**	0.689	0.005	1	500	0.9	new
GPT-2	57.756	**5.079**	**0.302**	0.672	0.101	20.503	0.698	0.002	1	500	0.9	old

Table 4: Mean of the evaluation metrics when changing the Wikipedia creation time

score, information density, and gunning fog all decreased compared to GPT-3. We hypothesize that, in particular in the setting of Wikipedia generation, it's because the penalty decreases the probability to generate repetitive entities and thus decreases the total number of generated entities. So, the generated text's complexity and informativeness would all decrease.

Old Wikipedia vs New Wikipedia To study the issue of data leakage, we also sample 3000 Wikipedia articles from older Wikipedia articles that the LLM might be trained on and compare the generated text with New-Wiki. We found that when we let GPT-2 perform generation on old Wikipedia articles, across all different parameter settings, the mean value of distinct n, essay score, Flesch reading score, information density, and S-BERT score increase slightly while the text complexity decreases. We believe it's because GPT-2 is trained on the corpus from the internet and it has seen some old Wikipedia before. And thus GPT-2 is able to store certain amount of knowledge into its parameters and thus would be able to generate text with higher quality. This verifies the data leakage issue and model memorization of large pre-trained language models (Elangovan, He, and Verspoor 2021).

Conclusions

This work provides a new evaluation framework for Wikipedia-style article generation. We propose the task of Wikipedia generation and provide a set of automatic well-rounded metrics to help researchers evaluate their generative language models' performance from several different perspectives. To do this evaluation without the issue of data leakage, we collect and release our New-Wiki dataset, which consists of Wikipedia articles created after GPT-3 is released as the test set. We then conduct an extensive evaluation of SOTA models including GPT-2, GPT-3, BLOOM, OPT, GLM, BART, and T5, and find interesting characteristics of these models and different parameter settings. In particular, we show and provide an explanation for 1) the large pre-trained language models are able to memorize knowledge into its parameters by comparing the generated text for old and new Wikipedia articles 2) what decoding methods and parameters would give the better model performance 3) GPT-3 is the well-rounded, highest scored model as of now compared to GPT-2, OPT, BLOOM, GLM, BART, and T5.

Limitations

1. Limited by computing resources, we did 10 completions for 100 prompts with OPT, GLM, and BLOOM. And thus some of the output may not be a perfect comparison with other model of which we did 20 completions on 2000 prompts. 2. OPT and BLOOM are released in 2022 and thus they possibly have seen our New-Wiki dataset which is sampled from June 2021 to December 2021. This might affect our evaluation results. 3. To make BART and T5 suitable for Wikipedia generation, we fine-tune them for 10 epochs on Wikipedia articles. There might be better fine-tuning strategies that can affect BART and T5's model performance.

References

Barrault, L.; Bojar, O.; Costa-Jussa, M. R.; Federmann, C.; Fishel, M.; and Graham, Y. 2019. Findings of the 2019 conference on machine translation (WMT19). Association for Computational Linguistics (ACL).

Berant, J.; Chou, A.; Frostig, R.; and Liang, P. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1533–1544.

BigScience. 2022. BLOOM. https://huggingface.co/bigscience/bloom. Accessed: 2022-08-14.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Caccia, M.; Caccia, L.; Fedus, W.; Larochelle, H.; Pineau, J.; and Charlin, L. 2018. Language GANs Falling Short. *CoRR*, abs/1811.02549.

Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; and Tang, J. 2021. GLM: General Language Model Pretraining with Autoregressive Blank Infilling.

Elangovan, A.; He, J.; and Verspoor, K. 2021. Memorization vs. Generalization : Quantifying Data Leakage in NLP Performance Evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 1325–1335. Online: Association for Computational Linguistics.

Gardent, C.; Shimorina, A.; Narayan, S.; and Perez-Beltrachini, L. 2017. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M.-W. 2020. Realm: Retrieval-augmented language model pretraining. *arXiv preprint arXiv:2002.08909*.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Karpinska, M.; Akoury, N.; and Iyyer, M. 2021. The Perils of Using Mechanical Turk to Evaluate Open-Ended Text Generation. *arXiv preprint arXiv:2109.06835*.

Khushali Thakkar, S. J. 2019. Project Title. https://github.com/sankalpjain99/Automatic-Essay-Scoring.

Kincaid, J. P.; Fishburne Jr, R. P.; Rogers, R. L.; and Chissom, B. S. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Ko, W.-J.; Durrett, G.; and Li, J. J. 2019. Domain agnostic real-valued specificity prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6610–6617.

Komatsuzaki, A. 2020. Current Limitations of Language Models: What You Need is Retrieval. *arXiv preprint arXiv:2009.06857*.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; Toutanova, K.; Jones, L.; Kelcey, M.; Chang, M.-W.; Dai, A. M.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7: 452–466.

Lapata, M.; Barzilay, R.; et al. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI*, volume 5, 1085–1090. Citeseer.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Roberts, A.; Raffel, C.; and Shazeer, N. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Roemmele, M.; Gordon, A. S.; and Swanson, R. 2017. Evaluating story generation systems using automated linguistic analyses. In *SIGKDD 2017 Workshop on Machine Learning for Creativity*, 13–17.

Sellam, T.; Das, D.; and Parikh, A. P. 2020. BLEURT: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Welleck, S.; Kulikov, I.; Roller, S.; Dinan, E.; Cho, K.; and Weston, J. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Wikipedia. 2022. Gunning fog index — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Gunning\%20fog\%20index&oldid=1067780465. [Online; accessed 15-August-2022].

Yuan, W.; Neubig, G.; and Liu, P. 2021. BARTScore: Evaluating Generated Text as Text Generation. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 27263–27277. Curran Associates, Inc.

Zellers, R.; Holtzman, A.; Rashkin, H.; Bisk, Y.; Farhadi, A.; Roesner, F.; and Choi, Y. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Appendix

The following table is the major experiments conducted. It evaluate 50 different models and corresponding parameter settings with our evaluation metrics. (Note BLEU score is under the scale of 100.)

Model	Flesch	Essay scoring	Entity Overlap	S-BERT	info_density	Gunning_fog	Distinct_n	Rep-n	BLEU	Top-p	Top-k	Temp	Wiki time
BLOOM	55.794	5.081	0.249	0.603	0.112	19.523	0.694	0.036	0.10	1	20	0.9	new
Bart	30.503	5.159	0.216	0.57	0.131	24.129	0.692	0.004	0.08	1	20	0.9	old
Fine-tuned GPT-3 Curie	50.193	5.121	0.242	0.628	0.142	22.661	0.661	0.006	0.09	1	20	0.9	new
GLM	50.812	5.157	0.291	0.692	0.122	19.882	0.543	0.208	0.06	1	20	0.9	old
GPT-3 Curie	49.032	5.134	0.746	0.76	0.15	22.892	0.655	0.009	0.08	1	20	0.9	new
GPT-3 Davinci	47.502	5.139	0.778	0.762	0.153	23.503	0.639	0.008	0.08	1	20	0.9	new
GPT-3 with penalty	50.161	5.119	0.242	0.628	0.142	22.689	0.661	0.006	0.09	1	20	0.9	new
OPT-66B	53.741	5.114	0.324	0.72	0.118	19.416	0.702	0.035	0.07	1	20	0.9	new
T5	57.491	5.012	0.227	0.669	0.108	18.937	0.676	0.01	0.16	1	50	0.9	new
GPT-2	53.064	4.899	0.631	0.689	0.114	17.552	0.654	0.782	0.09	0.5	50	0.1	new
GPT-2	56.235	4.929	0.566	0.69	0.116	16.094	0.662	0.726	0.10	0.5	50	0.5	new
GPT-2	60.038	4.977	0.436	0.694	0.119	15.46	0.68	0.573	0.09	0.5	50	0.9	new
GPT-2	53.637	4.931	0.651	0.691	0.116	17.447	0.656	0.797	0.11	0.5	50	0.1	old
GPT-2	57.731	4.941	0.591	0.691	0.117	15.66	0.672	0.742	0.12	0.5	50	0.5	old
GPT-2	61.521	4.993	0.479	0.694	0.118	15.275	0.69	0.598	0.12	0.5	50	0.9	old
GPT-2	54.631	4.917	0.608	0.691	0.116	16.816	0.658	0.759	0.14	0.7	50	0.1	new
GPT-2	58.781	4.946	0.521	0.692	0.117	15.259	0.671	0.685	0.12	0.7	50	0.5	new
GPT-2	62.099	5.024	0.338	0.699	0.116	15.927	0.688	0.298	0.11	0.7	50	0.9	new
GPT-2	55.794	4.937	0.628	0.692	0.117	16.637	0.659	0.775	0.10	0.7	50	0.1	old
GPT-2	55.651	4.924	0.596	0.691	0.116	16.298	0.66	0.751	0.12	0.9	50	0.1	new
GPT-2	61.066	4.97	0.457	0.693	0.118	14.906	0.676	0.596	0.11	0.9	50	0.5	new
GPT-2	60.711	5.046	0.304	0.701	0.108	18.295	0.685	0.035	0.10	0.9	50	0.9	new
GPT-2	56.835	4.938	0.613	0.693	0.118	16.213	0.662	0.764	0.09	0.9	50	0.1	old
GPT-2	61.889	4.978	0.491	0.695	0.117	14.59	0.685	0.612	0.13	0.9	50	0.5	old
GPT-2	61.651	5.061	0.354	0.7	0.107	18.024	0.695	0.035	0.14	0.9	50	0.9	old
GPT-2	55.249	4.92	0.593	0.692	0.116	16.355	0.656	0.748	0.11	0.95	50	0.1	new
GPT-2	61.277	4.979	0.432	0.694	0.118	14.892	0.679	0.55	0.11	0.95	50	0.5	new
GPT-2	60.127	5.05	0.296	0.698	0.106	18.858	0.684	0.018	0.12	0.95	50	0.9	new
GPT-2	57.292	4.941	0.614	0.694	0.116	16.132	0.663	0.762	0.08	0.95	50	0.1	old
GPT-2	62.621	4.989	0.467	0.695	0.117	14.533	0.687	0.565	0.14	0.95	50	0.5	old
GPT-2	60.987	5.062	0.350	0.699	0.105	18.607	0.693	0.017	0.11	0.95	50	0.9	old
GPT-2	56.125	4.925	0.591	0.691	0.116	16.154	0.66	0.746	0.12	1	20	0.1	new
GPT-2	61.668	4.988	0.407	0.697	0.117	15.03	0.679	0.479	0.13	1	20	0.5	new
GPT-2	60.456	5.048	0.300	0.702	0.106	18.563	0.679	0.021	0.12	1	20	0.9	new
GPT-2	57.228	4.939	0.61	0.692	0.117	15.926	0.664	0.759	0.11	1	20	0.1	old
GPT-2	56.000	4.926	0.590	0.691	0.116	16.244	0.659	0.746	0.11	1	50	0.1	new
GPT-2	61.869	4.993	0.397	0.696	0.118	15.081	0.683	0.459	0.10	1	50	0.5	new
GPT-2	59.574	5.051	0.288	0.695	0.103	19.311	0.683	0.01	0.13	1	50	0.9	new
GPT-2	57.342	4.946	0.608	0.694	0.115	16.244	0.663	0.76	0.11	1	50	0.1	old
GPT-2	62.533	5.003	0.435	0.697	0.116	14.773	0.69	0.473	0.08	1	50	0.5	old
GPT-2	60.274	5.063	0.339	0.694	0.103	19.175	0.692	0.008	0.09	1	50	0.9	old
GPT-2	55.558	4.927	0.589	0.691	0.116	16.252	0.66	0.745	0.11	1	100	0.1	new
GPT-2	61.681	4.991	0.393	0.696	0.118	15.17	0.682	0.455	0.07	1	100	0.5	new
GPT-2	58.568	5.053	0.279	0.688	0.102	19.875	0.684	0.007	0.16	1	100	0.9	new
GPT-2	57.429	4.942	0.609	0.692	0.117	15.895	0.664	0.758	0.12	1	100	0.1	old
GPT-2	62.667	5.005	0.435	0.696	0.117	14.77	0.69	0.467	0.15	1	100	0.5	old
GPT-2	59.289	5.068	0.328	0.688	0.102	19.651	0.693	0.005	0.08	1	100	0.9	old
GPT-2	55.92	4.926	0.59	0.691	0.116	16.263	0.659	0.745	0.14	1	500	0.1	new
GPT-2	61.765	4.993	0.392	0.695	0.117	15.091	0.683	0.449	0.09	1	500	0.5	new
GPT-2	56.956	5.065	0.259	0.675	0.102	20.685	0.689	0.005	0.12	1	500	0.9	new
GPT-2	56.897	4.939	0.61	0.692	0.117	16.062	0.663	0.76	0.11	1	500	0.1	old
GPT-2	57.756	5.079	0.302	0.672	0.101	20.503	0.698	0.002	0.10	1	500	0.9	old