Can Machines Tell Stories? A Comparative Study of Deep Neural Language Models and Metrics

AVISHA DAS AND RAKESH M. VERMA, (Member, IEEE)
Department of Computer Science, University of Houston, Houston, TX 77204, USA
Corresponding author: Rakesh M. Verma (rverma@uh.edu)

This work was supported in part by NSF under Grant DGE 1433817 and Grant CCF 1950297, and in part by the U.S. Army Research Office under Grant W911NF-16-1-0422 and Grant W911NF-20-1-0254.

ABSTRACT Massive textual content has enabled rapid advances in natural language modeling. The use of pre-trained deep neural language models has significantly improved natural language understanding tasks. However, the extent to which these systems can be applied to content generation is unclear. While a few informal studies have claimed that these models can generate ‘high quality’ readable content, there is no prior study on analyzing the generated content from these models based on sampling and fine-tuning hyperparameters. We conduct an in-depth comparison of several language models for open-ended story generation from given prompts. Using a diverse set of automated metrics, we compare the performance of transformer-based generative models – OpenAI’s GPT2 (pre-trained and fine-tuned) and Google’s pre-trained Transformer-XL and XLNet to human-written textual references. Studying inter-metric correlation along with metric ranking reveals interesting insights – the high correlation between the readability scores and word usage in the text. A study of the statistical significance and empirical evaluations between the scores (human and machine-generated) at higher sampling hyperparameter combinations ($t = \{0.75, 1.0\}, k = \{100, 150, 250\}$) reveal that the top pre-trained and fine-tuned models generated samples condition well on the prompt with an increased occurrence of unique and difficult words. The GPT2-medium model fine-tuned on the 1024 Byte-pair Encoding (BPE) tokenized version of the dataset along with pre-trained Transformer-XL models generated samples close to human written content on three metrics: prompt-based overlap, coherence, and variation in sentence length. A study of overall model stability and performance shows that fine-tuned GPT2 language models have the least deviation in metric scores from human performance.

INDEX TERMS Deep learning, transformer-based architecture, neural language models, natural language evaluation, story generation, generative pre-trained transformer (GPT), transformer-XL, XLNet, natural language generation.

I. INTRODUCTION

Natural language generation has gained popularity with new language resources and language models, which can be used to emulate the stylistic aspects of the training dataset. Beyond generating textual content such as stories and poems, language generation systems have been used for conversational dialog [17], automated headline generation [11], etc.

With the growing prominence of deep learning, an approach known as end-to-end learning [8] has become popular. Researchers have proposed several novel architectures capable of modeling robust representations of natural language – recurrent neural networks (RNNs) [47], sequential encoders-decoders (sequence-to-sequence learning) [8], generative adversarial networks (GANs) [9], and transformers with attention-modeling [51].

In recent years, use of large-scale neural language models trained on massive volumes of textual content has emerged as a solution to many natural language based tasks. Publicly available pre-trained models, such as OpenAI’s GPT [35], [36], AllenNLP’s ELMo [31], Google’s BERT [7], and Google/CMU’s XLNet [55], have improved performance on natural language understanding tasks considerably.¹

¹https://gluebenchmark.com/leaderboard
architectures have been pre-trained on massive\textsuperscript{2} amounts of raw or unlabeled textual content to build language models that can be readily applied to natural language based tasks with hardly any fine-tuning \cite{41} - a technique called “zero-shot” learning \cite{35}. Further fine-tuning these pre-trained models on a specific dataset (usually smaller in size than the original) leads to better models. This achieves better results than solely training a neural architecture on the new dataset, which results in an overfitted model \cite{41}, as was the case with previously proposed RNN-based generative systems.

In this paper, we examine the reproducibility and generalizability of multiple massively-trained language models in the realm of open-ended natural language generation. We study the behavior of these pre-trained models in a “zero-shot” setting and with fine-tuning on a dataset of human written stories and writing prompts. We also recognize that a selection of sampling hyperparameters (e.g., temperature, top-k value, etc.) for generation play an important role in determining the quality of the text generated. To evaluate the generated content, we use a range of metrics to compare with human written references – semantic relatedness, linguistic quality and syntactic style.

In addition, we explore three questions. Does fine-tuning a model on the dataset improve the quality of sample generation? What role do hyperparameters play in linguistic quality of generated text? Can writing quality be quantified using statistical measures? Are some metrics better at capturing the similarity/differences between the textual content generated by a machine or by a human? Our contributions are:

- We analyze correlations among several metrics.
- We present a rank-based evaluation of the metrics using a linear regularized model to see which metrics best distinguish the human writing and auto-generated instances.
- We analyze the results of the higher ranked metrics by varying the sampling hyperparameters (sampling parameter \( k \), softmax temperature \( t \)) for both pre-trained and fine-tuned models.
- We find combinations of \( k \), \( t \), models and metrics for which there is no statistically significant difference between the generated text and human-written content.

Our analysis of the metrics shows that, as expected, the story-prompt overlap percentages are highly correlated \((\rho > 0.9)\) at different values of \( n \) with bigram overlap being the best ranked metric. \( n \)-gram overlap metrics are highly correlated with the stylistic metrics – sentence length (mean, \( L_{\text{avg}} \) and standard deviation, \( L_{\text{sd}} \)) and noun distributions. Despite having low correlation between themselves and with coherency-based metrics, Dale-Chall readability score has a high positive correlation \((\rho > 0.9)\) with Type-token ratio, which shows the generation of unique and difficult words. The higher ranked metrics with positive correlation coefficient with respect to the text nature (human or machine-generated) are the bigram story-prompt overlap percentage, type-token ratio and standard deviation of sentence lengths.

Interestingly, we see that retraining the model on a subset of the domain specific data enhances model performance. Fine-tuning helps generate samples with story-prompt overlap closer to human writing, but the type token ratio increases with increasing sampling hyperparameter values. In our study, a model performs the best with respect to a metric when the scores of the generated stories are statistically similar to that of the human written references. However, as we will see, sampling parameters play an important role in text generation – in our analysis, we find that higher values of softmax temperature \( t \) (\( > 0.5 \)) and mid-range values of \( k \) (\( 50 < k < 500 \)) work the best. The models with samples performing at the human level on the top metrics are the fine-tuned GPT2 (117M and 355M) models along with the pre-trained models like GPT-110M and Transformer-XL. We see that the best hyperparameter combination is \( t = 0.75, k = 150 \) for generating samples closest to human writing. While the overlap of the generated stories with the prompt is the closest to human references at higher sampling parameters, variation in the sentence length of the textual content increases when compared to human references. Analyzing the \( L_{\text{sd}} \) shows that the deviation from the mean length increases with an increase in the sampling parameters \((t \text { and } k)\), with the pre-trained models performing closer to the human level.

\section*{A. PAPER ORGANIZATION}

Section II presents the related work. The dataset used for evaluation and the background required is described in Sections III and IV respectively. Section V describes the sampling and decoding algorithms used for generating the content. The experiment setup is in Section VI. The metrics, their correlation analysis and metric ranking are in Section VII. Results are in Sections X through VIII. Section XIV concludes.

\section*{II. RELATED WORKS}

A deep network trained on large amount of written text is capable of emulating the human writing style \cite{47}. We now summarize related work on language generation models and evaluation metrics.

\section*{A. NEURAL TEXT GENERATION}

We distinguish text generation models based on length of the generated content.

1) LONG-FORM CONTENT GENERATION

Automated long content generation is a difficult task – maintaining coherence becomes more challenging with an increase in the length. Deep neural architectures such as Recursive Neural Networks (RNNs), Long Short Term Memory (LSTM) networks are widely used for content generation \cite{14}, \cite{23}, \cite{47} owing to their ability to learn dependencies over long spans of text.

\footnote{the size of the training data is much larger than the size of the data used for the targeted task}
across the textual context [15]. LSTMs have been used for generating stories [8], Shakespearian Sonnets [54] and non-English poetry [12], [57], and for reading comprehension based tasks [24].

RNNs have also been used widely for sequence-to-sequence learning, e.g., [10], [19], [34], [59]. Standard sequence-to-sequence models have found application in open-ended content generation [48], but such straightforward encoder-decoder setups fail to generate content conditioned on a given starting seed. In [8], researchers proposed a state-of-the-art hierarchical neural fusion architecture using two seq2seq models [46] along with multi-scale gated attention mechanism ensuring relatedness of generated content to a given prompt. Since simple encoder-decoder architectures fail to model important meaningful representations of words and phrases, [21] used Gated Recurrent Unit (GRU)-based neural checklist models for recipe generation.

Other generation techniques include deep learning with Markov Models [52], variational auto-encoders [38], [44], and generative adversarial networks [33]. Researchers used multi-level variational auto encoders (sentence and word level) for generative decoder networks in [44]. The proposed system is used to generate Yelp reviews and abstracts of arXiv academic papers. Researchers in [10] use attention based encoder-decoder models for preserving coherence and context in the generated open-domain stories.

With the rise of language models pre-trained on massive amounts of textual content, generating content is becoming feasible [28]. Deep GPT2 language model [36] from OpenAI has gained a lot of attention in language generation applications. GPT models are built using a decoder-only transformer based architecture. The authors in [41] have compared generated content from the seq2seq fusion models and fine-tuned small GPT2 language model (117M parameters) with respect to story generation from writing prompts.

2) SHORT-FORM CONTENT GENERATION

While creative content (stories, poems) is longer text generation – automated systems are widely used for generating shorter textual content. These include generation of tweets with synthetic URLs [43] and reviews [56]. RNN-based methods for generating text messages are in [45]. The authors of [58] generate fake news articles from a given set of headlines using a transformer based architecture proposed by [35]. The authors found that readers preferred machine generated news articles as more readable than human written references.

Researchers in [2], [6], [13], ‘weaponize’ machine learning techniques to launch targeted attacks. The paper [2] uses a manual grammar-based approach for synthetic email generation and also studies the likelihood of a human to recognize a generated email from a legitimate counterpart. The system proposed in [6] uses an RNN-based architecture with word units for email generation. The authors train the model on a dataset of legitimate and phishing emails and generate samples by using greedy decoding techniques. However, the generated samples suffer from incoherence. For decoding, researchers in [41] use top-k sampling method. Previously researchers have used mostly greedy sampling methods for generating sequences [12], [14], [54].

B. NEURAL TEXT EVALUATION

We organize the literature on content evaluation metrics based on syntactic and semantic properties.

1) SYNTACTIC EVALUATION

Bangalore et al. [3] proposed automated accuracy-based metrics, which account for string matching as well as matches in the dependency-based parse tree to quantify the level of agreement between a given reference and the generated content. They also manually rate the generated content for quality and understandability using a scale of numeric scores from 1 (lowest) to 7 (highest). The paper [40] presents a comparison of automated and subject-based approaches for synthetic text with gold-standard reference texts using an NLG-based case study called ENIGMA. A Turing-style test for quality evaluation was also proposed in this paper. A number of grammar-based metrics: count of misspelled words, parsing score, and percentage of word overlap (BLEU) were compared with human evaluation results in [27].

2) SEMANTIC EVALUATION

Evaluating linguistic and semantic quality of generated text is essential. However, there could be a bias in choosing a metric or a method to evaluate the generated content automatically [50]. While existing automated evaluation metrics are not the best, manually evaluating generated text quality can be time consuming and prone to bias [27]. Authors put together a comprehensive evaluation of semantic-based automated metrics in [37]. To capture semantic relations across sentences at a word-level, the authors of [60] propose a similarity-based evaluation metric BERTScore. This metric is shown to perform better at measuring linguistic quality than existing metrics BLEU, ITER by correlating the score with human level judgements on two natural language understanding tasks: image captioning and machine translation.

Recently, researchers have looked into methods that compare text quality while taking into account hyperparameters that influence sample decoding from trained deep language models. The authors in [18] investigate how automated discriminators compare with human evaluators in this context. They also explore whether factors like sampling temperature and decoding parameters play a role in controlling the nature of the generated content. The paper [27] also reports results using a semantic similarity measure based on distributional similarity in text and Latent Semantic Analysis proposed by [16].

III. DATASET

We use the WritingPrompts dataset [8], [41] consisting of 303,358 pairs of prompts and manually written stories.
The dataset was collected by scraping three years of prompts and associated stories from Reddit’s WritingPrompts forum.

The dataset was built through crowdsourcing on an online community called WritingPrompts where users can submit premises to stories or prompts and can invite submissions from other online users. Each prompt or premise can have multiple story submissions, varying in length, topic and structure. The submitted story responses should follow or be inspired in some manner by the prompt. For other details about this dataset, we refer the readers to [8]. Table 1 shows an example of a story-prompt pair from the WritingPrompts dataset.

### Table 1. An example prompt-story pair from the WritingPrompts dataset.

| Prompt | Story |
|--------|-------|
| Single song word pool: “upward over the mountain” – iron & wine. I had an eye for the girl that night, mother she sold her body with some friends on the corner and I had money I saved for the weekend. she got in the car and I had taken her to where I lived upward over the mountain. I gave her my coat when we were in the garden. | Pick a singer and write a song/short story from only the words in their songs |

For evaluation, the dataset was split into three parts: training (90%), testing (5%) and validation (5%). Following the preprocessing steps in [8], the stories from the dataset are truncated to the first 1000 words for the experiments. For model fine-tuning, we preprocess the original dataset to create two new datasets using the Byte Pair Encoding (BPE) tokenization scheme – WritingPrompts-512 and WritingPrompts-1024, where the cut-off BPE length is 512 and 1024 tokens respectively. We discuss in detail the data preprocessing steps along with the dataset statistics in Section VI-A.

### IV. NEURAL GENERATIVE ARCHITECTURE AND LANGUAGE MODELS

While RNNs have been used widely for modeling contextual representations of textual content, such networks are computationally expensive [11] and fail to capture long term dependencies across longer sequences of written text. In this paper, we compare language models built by training transformer-based architectures with self-attention [51]. These include OpenAI’s GPT and GPT-2 architectures [35], [36]. Additionally, we discuss two transformer-based pre-trained generative language models: XLNet [55] and Transformer-XL [5], proposed by Google/CMU as modified versions of the GPT-2 architecture.

The network is built from a stack of encoders and decoders with self-attention, called the transformer blocks. The self-attention layer takes into account the importance of the neighbouring units in a given context piece as it encodes the input for boosting model performance. We refer readers to [51] and [36] for a detailed overview about self attention and transformer networks. To compare generative models with human writing, we look at the publicly available large-scale pre-trained language models released by OpenAI and Google. These models have yielded exemplary results in a wide variety of applications even when applied in a ‘zero-shot’ setting [36], [41]. While experimenting with pre-trained language models may be feasible for demonstrating a proof-of-the-concept application – an in-depth study and evaluation for a specific task would require model retraining or fine-tuning. We give the model names and their parameter sizes below.

#### A. OpenAI’s GPT

OpenAI’s GPT2 [35], [36] is essentially a large transformer-based network trained on web-scraped textual content. The generative GPT architecture is based on transformer decoder-only blocks. The largest trained model has 1.5 billion parameters and has been shown to outperform SOTA approaches on natural language understanding based tasks [36]. OpenAI’s GPT [35] model variants considered in this paper are the following – the smallest `openai-gpt`, and the subsequently released three GPT2 models [36], `gpt2-medium` and `gpt2-large`.

Fine-tuning refers to model retraining on a task and domain specific dataset without largely modifying the architecture, to further tune the model to the specific data to be evaluated on. However, fine-tuning such huge transformer models can be computationally intensive. See et al. [41] compares the fine-tuned version of the smallest GPT2 model (117 million parameters) with the Seq2Seq based Fusion model [8], SOTA architecture for open-ended story generation from writing prompts. In this study, we retrain two GPT2 models. After fine-tuning these models on each of the above datasets, we built the following four language models – `wp_512_117M`, `wP_1024_117M`, `wp_512_355M` and `wp_1024_355M`. We explain our fine-tuning experiment in Section VI.

#### B. GOOGLE/CMU’s TRANSFORMER-XL AND XLNet

Following the success of GPT2, Google/CMU released the Transformer-XL [5] and XLNet [55] models which improve upon the GPT2 language models. XLNet has been trained on multiple datasets which amount to a total of 136GB of text. Apart from the GPT models, models such as – XLNet [55] and Transformer-XL [5], have been shown to generate comparatively better textual content without prior training [1]. There are two pre-trained XLNet models – `xlnet-base-cased` and `xlnet-large-cased`, and the Transformer-XL model (`transfo-xl-wt103`). We refer the readers to the cited papers to explore these architectures in detail. However, at the time of our experiments, the authors had not released the aforementioned models for fine-tuning.

---

4 [www.reddit.com/r/WritingPrompts/](http://www.reddit.com/r/WritingPrompts/)

5 An experimental setup without any parameter or architecture modification and training

6 Dataset size is 40GB

7 [https://storage.googleapis.com/gpt-2/](https://storage.googleapis.com/gpt-2/)
V. SAMPLE GENERATION AND DECODING ALGORITHMS

Text generation using a trained language model is initiated by feeding one word at a time (starting seed word) to the model. The output is generated by stochastically selecting the most likely word to follow this given word from the probability distribution returned by the trained language model.

A. SAMPLING ALGORITHMS

Prior research uses techniques like random sampling, greedy sampling based and beam-search based decoding for sample generation. While greedy sampling techniques [12], [47], [54] choose the word with the highest probability from the distribution (argmax), this may not be the best solution for sample generation always. For sample generation, authors in [8], [36] use top-\(k\) random sampling scheme which is neither greedy (greedy decoding) nor non-deterministic (random sampling). The probability distribution at each timestep is redistributed among the top \(k\) tokens. The token with the maximum probability is chosen as the output.\(^8\) The papers [8], [36], [41] discuss the efficacy of this technique over conventional beam search and greedy methods [11] and experiment with different values of \(k\) for sample generation. In this paper, we use top-\(k\) sampling algorithm with varying values of \(k\).

B. SOFTMAX TEMPERATURE CONTROL

The final layer of the model, responsible for calculating the conditional probability, is a softmax normalization used for computing the distribution for the next word followed by subsequent sampling. We use temperature (\(\tau\)) as a hyperparameter for selecting word samples - regulating the parameter \(\tau\) in Equation 1 encourages or controls the diversity of the generated text. The novelty or eccentricity of the generative model can be evaluated by varying the temperature parameter between 0 < \(\tau\) ≤ 1.0. While, lower values of \(\tau\) generate relatively deterministic samples, higher values can make the process more stochastic. Equation 1 shows the probability distribution built by the model for the sequences of words along with the incorporation of temperature control.

\[
P(W_{t+1}|W_t) = \text{softmax}(W_t),
\]

\[
\text{softmax}(W_t) = \frac{w_j^t}{e^{-\frac{w_j^t}{\tau}}} \sum_{j=1}^{n} e^{-\frac{w_j^t}{\tau}}
\]

C. UNCONDITIONAL AND CONDITIONAL SAMPLING

Sample generation is the final step in complete text generative modeling. Common techniques to sample textual content using generative language models with or without retraining include two methods – unconditional and conditional sampling.

Generating samples unconditionally refers to using the generative model to output textual content without taking into account any user input. The model spits out text without taking actual starting seed or conditional statement from the user. Interactive conditional sample refers to generating samples based on user input. In other words, the user inputs some text and the trained language model does its best to fill in the rest. The command and parameters available is the same as that of unconditional sampling.

VI. EXPERIMENTAL SETUP

We describe the preprocessing steps followed to prepare the datasets for model fine-tuning and evaluation. Additionally, the section also describes how to acquire and setup the pre-trained language models during sample generation.

A. DATASET PREPROCESSING

Authors of [8] have provided a fairly clean and pre-processed version of the WritingPrompts dataset.\(^9\) The details of the original dataset has been described in Section III. Following [8] and [41], we truncate the human-written stories in the dataset to the first 1000 words. Table 2 summarizes the statistics of the truncated WritingPrompts dataset used in experiments.

| Dataset       | \#pairs | \(|V|_S\) | \(|V|_P\) | \(WC_S\) | \(WC_P\) | \(T_S\) | \(T_P\) |
|---------------|---------|---------|---------|---------|---------|--------|--------|
| Train         | 272,600 | 454,219 | 39,892  | 160M    | 7,7M    | 590    | 28.4   |
| Test          | 15,138  | 83,862  | 11,162  | 8.9M    | 423K    | 922    | 28.1   |
| Valid         | 15,620  | 85,576  | 11,129  | 7.1M    | 454K    | 585    | 29.0   |

\(\text{Byte Pair Encoding}\) (BPE) tokenization scheme was introduced by Sennrich et al. [42] as a data compression technique to improve machine translation tasks. The method reduces the total vocabulary size by keeping more frequent words while replacing the less frequent ones with a sequence of tokens. The BPE method ensures a balance between character- and word-level hybrid representations thus making it capable of effectively encoding large corpora. For the purpose of fine-tuning the language models, we create two additional datasets - WritingPrompts-512 and WritingPrompts-1024. These datasets are more suitable for the limited context.

\(^8\)https://huggingface.co/blog/how-to-generate#top-k-sampling

\(^9\)https://dl.fbaipublicfiles.com/fairseq/data/writingPrompts.tar.gz
size\(^\text{10}\) of the GPT2 models that will be fine-tuned on the prompt-story dataset.

The scheme is used to tokenize the \((\text{prompt}, \text{story})\) pairs and select the instances having a total token length equal to a certain given threshold (BPE token length). In this paper, the BPE token lengths chosen were 512 and 1024 respectively. Specifically, we keep the instances which have a total length\(^\text{11}\) less than or equal to 512 and 1024 BPE tokens respectively. The tokenization was done using the BPE model for English provided by the Python library ‘BPEmb’\(^\text{12}\). Tables 3 and 4 summarize the statistics of the resulting preprocessed datasets. Note that the WritingPrompts-1024 dataset is a better representative of the original dataset than WritingPrompts-512, as demonstrated by the descriptive statistics in Table 4.

### B. MODEL FINE-TUNING

Retraining the pretrained models is necessary to condition the model on the given set of story prompts and make the

---

\(^{10}\) the maximum BPE token size (here, 1024) that the model can process

\(^{11}\) length is calculated by concatenating the prompt and the story together

\(^{12}\) https://nlp.h-its.org/bpemb/#usage
retrained model generate stylistically and linguistically better stories from the prompts. The model fine-tuning can be termed similar to building a language model on the WritingPrompts dataset, where each prompt and story pair is regarded as one sequence separated by the delimiter token – |endoftext| . We use the Python implementation of the GPT2 models made available by OpenAI.\(^\text{13}\)

The fine-tuning experiment resulted in four models – the hyperparameters and the model training times and average loss achieved on the validation dataset are given in Table 5. The batch size and initial learning rate for the fine tuning experiments are chosen as 2 and 2 \(\times 10^{-5}\). The batch size and the learning rate\(^\text{14}\) were chosen based on the computation capability of our GPU. The models were trained using Python 3.6 on a Quadro P1000 GPU. For the four models, per-word perplexity using average loss (\(\text{loss}\)), i.e. if we consider it equal to \(e^{\text{loss}}\), falls in the range of 13.06 to 18.7 units which is lesser than the baseline fusion model mentioned earlier [8] on the validation datasets.

C. REPRODUCING PRE-TRAINED LANGUAGE MODELS

With the pre-trained Seq2Seq Fusion model [8] as baseline, authors in [41] fine-tune the small GPT2 model with 117M parameters (GPT2-117) [36] on a trimmed version of the WritingPrompts dataset (instances with 1024 BPE tokens). Besides model fine-tuning, we also apply the language models mentioned earlier for the story generation task in a ‘zero-shot’ setting. For reproducing the ‘massively’ pre-trained language models – OpenAI’s GPT and GPT2 and Google/CMU’s XLNet and Transformer-XL, we use the HuggingFace repository [53] which has implementations of these models for language generation.\(^\text{15}\)

\(^\text{13}\)https://github.com/nshepperd/gpt-2
\(^\text{14}\)Initial learning rate with an exponential decay and a decay rate of 0.96 and 10,000 steps
\(^\text{15}\)https://github.com/huggingface/transformers/tree/master/examples#language-generation
using the PyTorch 1.2.0 framework and Python 3.7.4 on a system with NVIDIA Tesla M10 GPU.

**D. SAMPLE GENERATION**

The stories in the test set of the WritingPrompts dataset are used as the references for comparing human written text (referred as **human** in our experiments) and auto-generated samples. The stories are generated using the top-\(k\) sampling technique [8], [36] during the generation process. The samples generated by the fine-tuned models are grouped together using the model names in Section VI-B.

For each of the pre-trained language models, two sets of experiments are performed to better visualize how the generated instances change with the hyperparameter tuning.

- Varying the softmax temperature parameter \(t\) \((t = \{0.25, 0.5, 0.75, 1.0\})\), while keeping the top-\(k\) sampling parameter \(k\) constant, where \(k \in \{0, 10, 50, 250, 1000\}\).
- Varying the top-\(k\) sampling parameter \(k\) \((k = \{5, 10, 50, 250, 500, 1000\})\), while keeping the softmax temperature parameter \(t\) constant, where \(t \in \{0.5, 0.75, 1.0\}\).

We generate samples by randomly selecting 200 different prompts from the test set. To compare performance with previous open-ended story generation literature, each generated story is limited to 150 words as mentioned in [8], [41]. For each of the selected test prompts, we randomly select a human-written story from the test set and use the first 150 words from the story for comparison during the evaluation step.

**E. PILOT STUDY**

OpenAI has made multiple GPT and GPT2 model variants publicly available. These language models vary in the number of trained parameters, size and number of layers, etc. [28] While [41] looks at a comparison of a fine-tuned GPT2 small (117M) model, there currently exists no literary work which compares all the provided language models - fine-tuned or pre-trained. This study looks at four fine-tuned models depending on the GPT2 model parameters and the BPE token size (see Section VI-B). Also, among the pre-trained GPT models provided by OpenAI, not all perform equally well.

Therefore, we conduct two sets of small scale pilot experiments to select the top two efficient pre-trained models among the pre-trained and fine-tuned GPT models. The two pilot studies are: (a) **Pilot Study I**: Comparing the pre-trained GPT language models; and (b) **Pilot Study II**: Comparing the fine-tuned language models.

For the set of pilot experiments, we choose to report the model performance using two sets of sub-experiments: (a) keeping temperature \(t\) = 1.0, we vary the value of \(k\) = \{5, 10, 50, 250, 500, 1000\}; and (b) keeping sampling parameter \(k\) = 50, we vary \(t\) = \{0.25, 0.5, 0.75, 1.0\}. The generated sample instances are compared with human references.

![FIGURE 5. Comparing all models for Bigram-Prompt overlap by varying \(k\) at constant \(t\) where (a) \(t = 0.5\). (b) \(t = 0.75\). (c) \(t = 1.0\).](image)

**VII. METRIC OVERVIEW, CORRELATION AND RANKING**

We give a brief overview of the evaluation metrics considered here. An important step is studying their pairwise correlations to filter out redundant metrics during analysis of system performance. Finally, we present a ranking of metrics with respect to their ability to distinguish between human-written and machine generated textual content.
FIGURE 6. Comparing all models for Bigram-Prompt overlap by varying $t$ at constant $k$ where (a) $k = 0$, (b) $k = 10$, (c) $k = 50$, (d) $k = 250$, (e) $k = 1000$.

A. METRIC OVERVIEW
We divide the set of metrics into five major groups based on their domain of evaluation – readability, syntactic style and complexity, part-of-speech usage, measure of coherence and prompt-based conditioning.

1) PROMPT-BASED CONDITIONING
The models must be able to condition well on the given story prompt, which means that the text generated must relate to a given initiating premise. To represent the conditioning capability of the model and how well they compare with human
FIGURE 7. Pilot Study I with pre-trained GPT models for \( L_{avg} \) by (a) varying top-\( k \) sampling parameter at constant temperature \( t = 1.0 \). (b) varying temperature \( t \) at a constant top-\( k = 50 \).

FIGURE 8. Pilot Study II with fine-tuned GPT models for \( L_{avg} \) by (a) varying top-\( k \) sampling parameter at constant temperature \( t = 1.0 \). (b) varying temperature \( t \) at a constant top-\( k = 50 \).

references – we look at \( n \)-gram \((n \in \{1, 2, 3\})\) word overlap between the prompt and the generated story with stopword elimination. However, these models often repeat terms from the given prompt, which could lead to higher overlap. But the main indicator should be how the generative models perform with respect to the overlap percentage observed in human writing.  

2) STYLE AND COMPLEXITY
While there is no good way to measure stylistic complexity of textual content, an overly complex piece of text can reduce readability while poorly written content demonstrates lack of sophistication [41]. Along with considering the mean \( (L_{avg}) \) and standard deviation \( (L_{sd}) \) of the sentence length in the stories, we study the type token ratio \( (as \text{percentage}, \ ttr\_pc) \) to observe how the stylistic quality of the generated text compares to human writing.

3) READABILITY MEASURES
Readability metrics are an automatic and easy measurement of text difficulty [20], [27]. Readability scores like Flesch Reading Ease \( (fre) \) [22] and Dale-Chall Readability score \( (dcr) \) [32] attempt to quantify the level of difficulty of a text with respect to the reader’s education level. While in FRE, text complexity is calculated using the average length of the sentence as well as presence of polysyllabic words; the DCR score takes into account familiarity or knowledge of a word while calculating readability. We discuss more in the experimental results section.

4) MEASURE OF COHERENCE
It is important to evaluate the coherence in written content to determine whether there is correlation across the units (sentence or words). We propose the sentence connectedness...
A. Das, R. M. Verma: Can Machines Tell Stories? A Comparative Study of Deep Neural Language Models and Metrics

FIGURE 9. Comparing all models for L_avg by varying k at constant t where (a) t = 0.5. (b) t = 0.75. (c) t = 1.0.

5) PART-OF-SPEECH USAGE
Part-of-Speech (POS) usage can be a useful indicator of linguistic quality. An exploratory analysis of the textual content in existing research reveals that Noun and Verb tags are the most commonly occurring parts of speech [39]. Hence we primarily compare the frequency distributions of these two tags between the synthetic and human written references.

B. METRIC CORRELATION
We use the mean metric scores on the generated instances across different hyperparameter combinations to compute the correlation between each pairwise metric. The metrics are given ranks on the basis of statistical scores based on the metrics’ correlation with the outcome variable (here, ‘label’) as well as each other. We use Pearson’s Correlation Coefficient ($\rho$) for this purpose.

Figure 1 is a heatmap showing the measure of correlation amongst the metrics as well as the outcome variable. Note the low level of correlation between the readability metrics (fre and dc). As expected, there is a high positive correlation ($\rho > 0.9$) among the story-prompt n-gram overlap metrics (uniOL, biOL and triOL). Interestingly, these overlap metrics have relatively high positive correlation ($\rho > 0.75$) with the style metrics: average sentence length and the distribution of nouns in the textual content. Other interesting high positive correlations are between Dale-Chall Readability score and Type-Token Ratio, and between average sentence length and noun usage. A high correlation between the Dale-Chall Readability score and Type-token ratio shows that the generated content has an increased occurrence of difficult and unique words. Below, we do regression analysis of these metrics with the outcome variable.

C. METRIC RANKING
In this experiment, we use the metric values for each story and the label ‘human’ versus ‘automatic’ for the generated samples. The scores of the evaluation metrics are given to the well-known regression-based LASSO algorithm [49]. In this way, we can determine which metrics are better at distinguishing between manual and auto-generated samples. LASSO applies a penalty based technique that determines how many “features” are retained. Additionally, using cross-validation to choose the penalty factor ($\alpha$) helps improve model generalizability and select the best fitted model. Here, we use 5-fold cross-validation with the linear ‘LassoCV’ model provided by Python’s Scikit-Learn package. The absolute value of the correlation coefficient determines the level of impact of a unit change in the known variable (here, the evaluation metrics) on the estimated

(sent_conn) metric to evaluate the cohesion at the sentence level in textual content. Using Sent2Vec-based embeddings, we transform each sentence in the story to their embedding vectors. To capture the difference in sentence meaning, we calculate the angular variation in consecutive sentence vectors in radians. We compute the standard deviation of the list of pairwise angular differences obtained for each adjacent sentence pair. Lower the variation, more is the connectedness or coherence across the text content.
variable (here, the nature of written content – human or machine) and the sign (positive or negative) of the coefficient determines the nature of the impact of the variable. The model iterates over a 100 possible alpha values to select the best $\alpha$. For this work, the best $\alpha$ is 0.000651 and the best model score (coefficient of determination, $R^2$) is 0.840436.

We see that the metrics capable of giving the most important information for distinguishing the generated samples from human references are $N$-gram overlap percentages between the story and the prompt, specifically bigrams and unigrams. The bigram overlap shows a strong positive coefficient of correlation while unigram overlap percentage has

---

**FIGURE 10.** Comparing all models for $L_{avg}$ by varying $t$ at constant $k$ where (a) $k = 0$. (b) $k = 10$. (c) $k = 50$. (d) $k = 250$. (e) $k = 1000$. 
a similar but negative correlation with the outcome. Here $Y$ denotes the outcome variable i.e. ‘label’. Dale-Chall readability score and Type-token ratio percentage also appear as metrics with strong coefficients for distinguishing between the human and non-human content. The statistical properties of number of words in the textual content at the sentence level like mean sentence length ($L_{\text{avg}}$) and standard deviation of sentence length ($L_{\text{sd}}$) are also highly correlated with the label. While mean sentence length has a negative correlation coefficient, standard deviation has a similar but positive correlation with the nature of the content. The measure of interaction of the metrics with the outcome variable in terms of their coefficients based on the fitted linear Lasso model is given by Equation 2:

$$Y = 3.219 \times biOL + (-2.704) \times uniOL + (-1.26) \times triOL + (-1.018) \times dc + (-0.979) \times L_{\text{avg}} + (0.887) \times L_{\text{sd}} + 0.574 \times ttr_{pc} + (-0.088) \times sent_{conn} + (-0.069) \times verb + (-0.029 \times noun) + (-0.023) \times fre$$

Although, the model does not predict any metrics with zero coefficient, we see from Figure 2, the lowest coefficient values are assigned to the metrics $fre$, $sent_{conn}$, $noun$ and $verb$ frequency distributions. The other metrics with lower power of distinguishing between human and non-human (generated) writing are the percentages of noun and verb usage. While $L_{\text{avg}}$ and $L_{\text{sd}}$ may not be the top ranked parameters because of variance in sentence lengths in human and machine-written content. Also, while $L_{\text{sd}}$ is positively correlated to the label ($Y$), $L_{\text{avg}}$ has a negative correlation coefficient.

We now present the metric-wise performance of the models. All experiments compare the metric scores of the generated model samples in two sets of sub-experiments: (i) varying $k$ at different constant values of temperature $t$; and (ii) varying $t$ at different constant values of $k$.

**VIII. PROMPT-BASED CONDITIONING**

Conventional models often fail to produce text that is semantically and contextually related to a given prompt/seed [8].

181270
Conditioning on the provided starting seed acts as a guide for the generative model by providing some prior context for it to choose the best possible sequence of words and/or phrases from the distribution. A measure of sample-prompt relatedness ideally acts as an indicator to how the generative language model can ‘stick’ to the context in the given seed. A higher score of overlap (close to 100%) can also be due to samples containing only words from the prompt repeatedly. So our targeted level of bigram overlap percentage is 0.9% in the human written samples – thus humans writes samples which reuse words from the given prompt very scarcely.

Using the Python NLTK toolkit [25], we look into the percentage overlap [41] of uni-, bi-, and tri-grams between the generated stories and the prompt. The analysis above shows the feature importance of bigram overlap – the percentage is a good indicator of how the human references differ from human samples. We also note from prior correlation analysis (Figure 1), that there exists a very strong correlation amongst \( n \)-gram overlap percentages. In this section, we focus on the findings from the bigram overlap percentage results.

Table 6 show the statistical significance of the bigram overlap percentage scores of generated text with the human references. We see that not many pre-trained models can generate samples which are equivalent in overlap scores with the human references. Next, we look at how these models actually perform with different settings of the sampling hyperparameters.

**TABLE 6.** \((k, t)\) combinations of pre-trained models where Bigram Overlap Percentage is statistically similar to human references.

| \(k\)  | \(t\) | 0.25 | 0.5 | 0.75 | 1.0 |
|-------|-------|------|-----|-----|-----|
| 0     | TX    | G2, OL, w10-1M |
| 10    | TX    | G2, OL, w5-1M |
| 50    | TX, OG, w5-1M | TX, w5-1M |
| 1000  | OG, w10-1M | w10-1M, w5-1M |

Among the pre-trained GPT2 models, the bigram overlap for the models, OG, G2 are the closest to the human level. Figures 3a and 3b show a decreasing trend in the overlap percentage with an increase in the hyperparameter values with varying \( k \) and \( t \). At \( k \) values of 250, 500 and 1000 at \( t = 1.0 \), OG and GPT2 are the closest to the human levels. This explains the lower overlap percentages of the models for \( k = 50 \) with varying \( t \). A similar trend is shown in Figures 4a and 4b for the samples generated by the fine-tuned models. At \( t = 1.0 \), and at values of \( k \) greater than 100, the fine-tuned models w10-1M and w5-1M score the closest to the human bigram overlap percentages. Therefore, the overlap percentages shown in Figure 4b for \( k = 50 \) are not significantly closer to the human references for the model samples.

Taking the best generative models from the pilot experiments, we look at the trend in the percentage of bigrams common to the story and the prompt. With changing the sampling parameter \( k \) at constant temperature, the Figures 5a, 5b and 5c show how the bigram overlap percentage changes. The same for varying softmax temperature \( t \) are shown in Figures 6a, 6b, 6c, 6d and 6e.

A higher overlap percentage means that generator is repeating words from the prompt, as seen in the samples from the pre-trained models at lower values of hyperparameters. Among the pre-trained models, XLNet (XB and XL)
larger GPT2 models (GM and GL) generate samples that are not close to the human references. However, the pre-trained TX and smaller GPT models OG and G2 generate content having minimal overlap (approx. 1%) with the story prompts at temperature $0.75$ and $k$ values greater than 150 (Figure 5b). This is a sign that the models maybe generating unique words.

Key observations on bigram overlap are:
- The pre-trained and fine-tuned models generate samples with lower overlap bigram percentages (approx. 1%) with an increasing value of the sampling $k$ and $t$ values. The percentages are closest to the human level at $t = 0.75$, 1.0 and $k = 500$, 1000.

**FIGURE 14.** Comparing all models for $L_{sd}$ by varying $t$ at constant $k$ where (a) $k = 0$. (b) $k = 10$. (c) $k = 50$. (d) $k = 250$ (e) $k = 1000$. 
A. Das, R. M. Verma: Can Machines Tell Stories? A Comparative Study of Deep Neural Language Models and Metrics

FIGURE 15. Pilot Study I with pre-trained GPT models for Type Token Ratio (TTR) percentage by (a) varying top-$k$ sampling parameter at constant temperature $t = 1.0$. (b) varying temperature $t$ at a constant top-$k = 50$.

FIGURE 16. Pilot Study II with fine-tuned GPT models for Type Token Ratio (TTR) percentage by (a) varying top-$k$ sampling parameter at constant temperature $t = 1.0$. (b) varying temperature $t$ at a constant top-$k = 50$.

• The pre-trained models generating the most human-like samples are OG, G2 and TX, and the fine-tuned models are w10-1M and w5-1M.
• Lower values of sampling parameters have higher overlap with the prompt for the pre-trained models. However the samples generated by fine-tuned models at lower values of sampling parameters ($t = 0.5$ and $k = 10$) show low prompt-overlap percentages, which are very similar to the overlap scores of human authored text.

IX. SYNTACTIC STYLE AND COMPLEXITY

Now, we observe the syntactic quality of generated content using sentence length and type token ratio.

A. SENTENCE LENGTH

Sentence length has been used in previous research to estimate the level of syntactic complexity [22], [39], [41]. The authors in [39] consider average sentence length a reliable metric that can capture text genre and overall content readability. Although the feature ranks lower in our metric ranking analysis (Section VII), we include this metric to spot any interesting trends. Table 7a shows that the pre-trained OG generates samples that are similar to the human written references at different values of $k$ at different constant $t$. The table reveals that the fine-tuned w10-3M and w5-3M perform the best compared to human scores.

Figure 7 shows OG performing similar to human writing – the best results at $k = 50$, $t = 0.75$. The $L_{avg}$ values for the other GPT2 models are significantly higher than the human level, thus showing they generate longer sentences. The observation also supports the statistical significance study for $L_{avg}$. The results of PS-II are shown in Figures 8a and 8b. The $L_{avg}$ values of the sentences generated by models w5-1M, w5-3M and w10-3M are close to human scores. It is interesting to note that at $t = 1.0$ and $k = 10$, the models w10-3M and w5-1M have scores exactly equal to the human reference. $L_{avg}$ increases with an increasing $k$. 
FIGURE 17. Comparing all models for Type Token Ratio (TTR) percentage by varying $k$ at constant $t$ where (a) $t = 0.5$. (b) $t = 0.75$. (c) $t = 1.0$.

We consider $L_{avg}$ of the generated content of the top models from PS-I and PS-II and the samples from XLNet and Transformer-XL models at varying combinations of $(k, t)$. For different constant values of $t$, the trends with varying $k$ are shown in Figures 9a, 9b and 9c. While varying the temperature $t$, the Figures 10a, 10b, 10c, 10d and 10e show the changes in $L_{avg}$ at constant values of $k$. We observe that the change in $L_{avg}$ is uniform with the change in the corresponding sampling parameter in both the cases. The models that generate samples with $L_{avg}$ closest to human references are – fine-tuned $w5-1M$ and $w10-3M$ and pre-trained OG. However, the XLNet models perform poorly generating sentences with much higher lengths. The best set of sampling parameter combination is $k = 1000$ at $t = 0.75$.

We further analyze the standard deviation in the sentence length ($L_{sd}$) of the generated textual content and how the metric score varies with respect to change in the model nature and sampling parameters ($t$ and $k$). The results of the pilot studies for the pre-trained and fine-tuned GPT models are shown in Figures 11 and 12 respectively. From the results, we choose the pre-trained models – OG and GM and fine-tuned models – $w5-3M$ and $w10-3M$ as the best models for or further comparison with the XLNet based models - XL, XB and TX. We compare the pre-trained and selected fine-tuned models based on the standard deviation of the length of generated sentences to that of the human authored content. The results of the metric scores with varying $k$ at different values of constant $t$ is shown in Figure 13. We observe that the $L_{sd}$ in general has a more stable trend for all the models except the XLNet-based models. The trends are closest to the human scores at the $t = 0.75$ showing that most models perform closest to the human level at $t = 0.5, 0.75$. There is a deviation in the overall $L_{sd}$ scores at $t = 1.0$ with varying $k$ values. Somewhat similar observations can be drawn from Figure 14 with varying $t$ values at constant $k$.

Observations on mean and standard deviation of sentence length are:

- $L_{avg}$ does not show a uniform or consistent trend with varying sampling hyperparameters for the GPT2-based pre-trained models as seen in our PS-I analyses. The fine-tuned models perform similar to human written content on this metric.
- XLNet models underperform on this metric; the best model is surprisingly OpenAI’s GPT model – the smallest transformer-based GPT variant along with Google/CMU’s Transformer-XL model.
- Fine-tuned models perform the best – GPT2-medium model trained on WritingPrompts-1024 dataset performs
FIGURE 18. Comparing all models for Type Token Ratio (TTR) percentage by varying $t$ at constant $k$ where (a) $k = 0$. (b) $k = 10$. (c) $k = 50$. (d) $k = 250$. (e) $k = 1000$.

the best (Table 7a) at ($k = 1000$, $t = 0.75$). For the model, wP\_512\_117M, the best combination is ($k = 50$, $t = 0.75$). Models perform better at moderate temperature values (0.75) and higher values of $k$.

B. TYPE TOKEN RATIO

Type-token ratio (TTR) [39] measures complexity, lexical richness or variety in vocabulary. TTR is the ratio between the total vocabulary\footnote{number of unique words} or types, and the total number of words or tokens in a given piece of textual content. Higher TTR value indicates greater lexical richness of the text. Similar to the other sections, Table 7c reports the statistical significance study of TTR for auto-generated samples with respect to human references. The majority of the models
occur at \( t = 1.0 \) for different values of \( k \). Content generated by pre-trained larger GPT2 based models (\( \text{GM} \) and \( \text{GL} \)) and \( \text{TX} \) have scores similar to human references. Among the fine-tuned models, the top ones are \( \text{w10-1M} \) and \( \text{w5-1M} \). We notice that at a higher \( k \), the samples generated by the models digress from the human reference score – maybe due to the generation of more unique words than present in human writing. The mean TTR percentage recorded by human writing is 40% for the WritingPrompts dataset.

The pilot studies show a comparison of pre-trained and fine-tuned GPT2 models for varying \( k \) and \( t \) parameters – the studies reveal the generation of more unique words with an increasing parameter value (\( k \) or \( t \)). This may indicate that the model is generating random strings of unique words. The samples from models \( \text{OG} \), \( \text{GM} \) and \( \text{GL} \) perform closest to the human references at \( \{ t = 0.75, k = 50 \} \) and \( \{ t = 1.0, k = 10 \} \) as seen in Figures 15a and 15b. For PS-II, the TTR values of the fine-tuned models are shown in Figures 16a and 16b reveal the same increasing trend. The models \( \text{w10-1M} \) and \( \text{w5-1M} \) perform the best at \( \{ t = 0.75, k = 50 \} \) and \( \{ t = 1.0, k = \{10, 0\} \} \). For the comprehensive model study, we look at the metric scores for samples generated at different combinations of \( k \) and \( t \). Figure 17b shows that the pre-trained models \( \text{OG} \) and \( \text{GL} \) generate samples with TTR scores closest to human writing along with the fine-tuned model \( \text{w10-1M} \) at higher values of \( k \). For \( t = 1.0 \), the samples generated by the models generate human level samples at lower \( k \) values of 0 and 10 (Figure 17c).

Figure 18a shows the XB model samples have values close to human reference level. The models \( \text{TX} \), \( \text{OG} \) and \( \text{GM} \) have values comparable to humans at \( t = 0.75 \). A similar observation can be made from Figure 18c.

Key takeaway points on type-token ratio are:

- TTR scores of the generated instances are close to the human level at lower values of \( k \) i.e., \( k = \{0, 10\} \) at \( k = \{50, 100, 150, 1000\} \)
FIGURE 21. Comparing all models for Flesch Reading Ease (FRE) scores by varying $k$ at constant $t$ where (a) $t = 0.5$. (b) $t = 0.75$. (c) $t = 1.0$.

A. Das, R. M. Verma: Can Machines Tell Stories? A Comparative Study of Deep Neural Language Models and Metrics

**X. READABILITY MEASURES**

We use Python’s *textstat* [4] library to calculate the Flesch reading ease (*fre*) [22] and Dale-Chall Readability score (*dcr*) [32], for varying $t$ and $k$ values. We report the models’ performance on both set of metrics since there exists a low correlation between the two metrics.

### A. FLESCH READING EASE

For narrowing down the models and the best set of sampling hyperparameter ($k$ and $t$) combinations, we look at the t-statistic and p-value given by the one-sample T-test of statistical significance.\(^{20}\) The aim is to find the fine-tuned and pre-trained model(s) and ($k$, $t$) where the generated samples have a mean FRE value that is statistically similar to that of the human written references (approx. 64.7).\(^{21}\) The results of this analysis is shown in Table 8. We observe that the FRE metric is not good for differentiating between human writing and synthetic examples, that is consistent with the LASSO results in Section VII. A closer look at the evaluation experiments provides a better insight.

For the pilot studies, Figures 19a and 19b show the results for PS-I for varying $k$ and $t$ respectively, and Figures 20a and 20b show the same for PS-II. Figure 19b shows that the model GL generates samples with values comparable to human written counterparts at $t = 1.0$. Additionally, the FRE scores of the G2 model reflected in Figure 19a, with varying $k$ at $t = 1$, are closer to the human written scores. This figure shows a more stable trend for models GM and GL as well. However, we see some sudden drops in the score for specific parameter combinations – GM has a very low value (3.68096) for $k = 250$ at $t = 1.0$.

---

\(^{20}\)https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.ttest_1samp.html

\(^{21}\)https://readabilityformulas.com/flesch-reading-ease-readability-formula.php

---

**TABLE 8. ($k$, $t$) combinations of pre-trained models where FRE is statistically insignificant from human references.**

| $k$ | $t$  | FRE score |
|-----|------|-----------|
| 0   | 0.5  | 64.7      |
| 0   | 0.75 | 64.7      |
| 1   | 0.5  | 64.7      |
| 250 | 1.0  | 64.7      |

---

size. Thus at higher $k$ values, the models tend to generate more random unique words than present in human writing.

- The pre-trained models OG, TX and the fine-tuned w10-1M shows TTR scores closest to human level with varying $k$ at higher $t$ values ($t = \{1.0, 0.75\}$). The models seem to perform better at lower $k$ values with respect to their TTR scores.

- Varying the temperature $t$ at constant values of $k$ shows a different trend in the model samples – the pre-trained XL and XB perform closer to the human reference levels and have a consistent trend with change in $t$.
sheds light on the top performing fine-tuned models using a similar approach. Figure 20a shows a much smoother trend or variability in the FRE values with changing $k$. The change in softmax temperature $t$ at $k = 50$ in Figure 20b, shows a steady increase in the FRE value with the model $w_{10-3M}$ being the top performer, while $w_{5-3M}$ can be called a second best consistent model based on the trend variability. Again, our observations from PS-II can be supported by the results shown in Table 8 – where $w_{5-3M}$ and $w_{10-3M}$ occur most frequently.

Finally, we combine the selected pre-trained and fine-tuned GPT models with the set of pre-trained transformer
models from Google/CMU - TX and XL and XB. Changing the top-\(k\) sampling parameter at different values of constant softmax temperature \(t\) reveal some interesting insights into model performance with sampling parameter tuning. While the models XB and XL consistently perform poorly in all three scenarios shown in Figures 21a, 21b, the samples from the pre-trained transfo-xl model achieves a consistent FRE scores close to the human references. The fine-tuned GPT2 models, w10-3M and w5-3M also generate textual content with readability scores similar to human written textual content at a combination of \(t = 0.5\) and \(k = 500\). But this is expected since both these models have been trained additionally on a preprocessed version of the provided WritingPrompts dataset. These fine-tuned models along with the pre-trained TX and G2 (small and medium) also perform well for \(k = \{10, 50, 150, 250, 1000\}\) at \(t = 0.75\) as seen from Figure 21b. For varying \(t\), we observe that the samples generated at a top-\(k\) sampling value of \(k = 10\) are the most consistent and closest to the human reference scores for models OG, TX and XL. But, on average the worst scores (negative FRE) are observed by samples generated by the XB and XL. An interesting observation is that we see higher overlap at \(t = 0.5\) and \(t = 0.75\) of the sample scores with the human references for the above mentioned.

Findings from the analysis of FRE scores are:

- The experiments and statistical significance results support that samples generated by the pre-trained models openai-gpt, transfo-xl and xlnet-large are closest to the human samples in FRE scores.
- The fine-tuned GPT2-medium model trained on the WP-1024 (wP_1024_355M) dataset generate samples with a reading ease score similar to the human references.
- The models perform better on this metric at top-\(k\) values of \(k = 10, 50, 1000\) and softmax temperatures of \(t = 0.75, 1.0\).
- Contrary to the previous automated metric study [27], FRE does not capture the linguistic quality of the generated content well. The correlation (Figure 1) and the metric ranking (Figure 2) along with high number
A. Das, R. M. Verma: Can Machines Tell Stories? A Comparative Study of Deep Neural Language Models and Metrics

**FIGURE 25.** Comparing all models for Dale-Chall Readability (DCR) scores by varying $k$ at constant $t$ where (a) $t = 0.5$. (b) $t = 0.75$. (c) $t = 1.0$.

of statistically insignificant (Table 8) hits support that Flesch Reading Ease is not well suited to distinguish between generated and human authored content.

**B. DALE-CHALL READABILITY SCORE**

First, we perform a test of statistical significance using the one-sampled t-test. The results for the pre-trained and fine-tuned models are shown in Table 9. We see that among the pre-trained models, TX has generated samples similar in DCR scores to human writings. An important observation here is that the fine-tuned models perform the best at $t = 1$ and $k = 1000$, that is almost the entire length of the provided human stories used for training.

For evaluation, we start with identifying the best GPT2 models through Pilot Study I (PS-I) for pre-trained and Pilot Study II (PS-II) for fine-tuned GPT2 models. Figures 23a and 23b show the results for PS-I – samples from models G2, GM and GL perform best. However, at temperature $t = 1.0$, the scores of the samples at $k = 50$ in Figure 23a are the closest to the human preferences (little less than 6.0) that can also be seen in the variability of the values generated by the models G2 and GM. Although, OG and GL may have statistically similar scores to human references at particular values $k$ and $t$, a closer look shows more homogeneity in the plots generated by the models G2 and GM. Figures 24a and 24b show a similar upward trend for PS-II as seen in PS-I with an increase in the sampling parameter value depending on the setup. Interestingly, the models w10-1M, w5-3M and w10-3M generate the samples that have DCR scores closest to human references at $k = \{250, 500, 1000\}$. The top two models recording scores similar to the human written stories in Figure 24b are w5-3M and w10-3M. This is supported by the statistical significance values in Table 9, where we see more overlap between the GPT2 models, GL, w5-3M, w10-3M and w5-1M, at higher $t$ and $k$ values. Taking the top models from PS-I and PS-II, we compare the stories generated by the following models – gpt2 and gpt2-medium from PS-I; wP_512_355M and wP_1024_355M from PS-II; xlnet-base-cased, xlnet-large-cased and transfo-xl with the human references. Like the previous experiments, this analysis also has two sub-experiments: (a) varying $k$ at different $t$ values as shown in Figures 25a, 25b and 25c respectively; and (b) varying $t$ at different values of $k$ shown in Figures 26a, 26b, 26c, 26d and 26e. Looking at DCR scores with varying $k$ at constant $t$, transfo-xl generates better quality samples scoring closer to human writing. For the other models, scores are lower when the top-$k$ sampling is done at $t = 0.5$ and the values increase slightly when $t = 0.75$ as observed from the change in gradient of score plot. This increasing gradient is even more apparent in Figure 25c, with the samples generated by models gpt2, gpt2-medium. Fine-tuned wP_512_355M

| $k$ | $t$ | TX | OG | TX | GL | w5-1M, w10-1M, w10-3M |
|-----|-----|----|----|----|----|---------------------|
| 0   | 0.25| TX | OG | TX | GL | w5-1M, w10-1M, w10-3M |
| 10  | 0.75| TX | GL | w5-1M, w10-1M, w10-3M |
| 250 | 1.0 | GL | w5-1M, w10-1M, w10-3M |
| 1000| 1.0 | w5-1M, w10-1M, w10-3M |

**TABLE 9.** $(k, t)$ combinations of pre-trained models where DCR is statistically insignificant from human references.
and wP\_1024\_355M score close to human references. For \( k = 1000 \), the fine-tuned models generate samples with scores closer to human references at \( t = 1.0 \). The best results are observed at \( k = 250 \) for the following models – gpt2, gpt2-medium and fine-tuned wP\_512\_355M and wP\_1024\_355M – which additionally perform well in terms of generating samples as readable as the human references.

Key takeaway points for this metric are below:
- The overlap of the metric scores at higher values of temperature and sampling value \( k \). For \( k = 1000 \), the fine-tuned models generate samples with scores closer to human references at \( t = 1.0 \).
- The best results are observed at \( k = 250 \) for the following models – gpt2, gpt2-medium and fine-tuned
wP_512_355M and wP_1024_355M – which additionally perform well in terms of generating samples as readable as the human references. The model transo-xl also perform well in this set of evaluation.

• The best combinations are \( t \in \{1.0, 0.75\} \) and \( k \in \{250, 500, 1000\} \). This is corroborated with the evidence of statistical significance from the experiments in Table 9.

• DCR is a good metric for differentiating between the human written and machine written samples.

XI. SENTENCE CONNECTEDNESS

One important aspect of evaluating generated content with respect to the human written counterparts is looking at the coherence. We propose a metric to measure the connectedness of sentences. Sentence embeddings capture the semantic nature and overall collective dependency between the words as a whole much better than word-based embeddings like Word2Vec [26] and GloVe [30]. Therefore, each sentence in the story along with the provided prompt is converted into a sentence-based embedding vector [29].

We use the Sent2Vec Python library\(^{22}\) proposed in [29] and the pre-trained Wikipedia Bigram model\(^{23}\) for the 700 dimensional sentence vectors. For each story, we compute the angle between each pair of consecutive sentences. The angular difference between two consecutive equi-length (700 dimensional) sentence vectors is calculated in radians. Finally, to capture the variation in the angular difference, we compute the standard deviation of the list of pairwise angular differences obtained for each adjacent sentence pair. The deviation in the sentence connectedness values in human writing is close to 0.21. We hypothesize that a greater value of standard deviation in the angular difference is an indicator of greater variability in the sentence coherency. The statistical

\(^{22}\)https://github.com/epfml/sent2vec

\(^{23}\)https://github.com/epfml/sent2vec#downloading-sent2vec-pre-trained-models
A. Das, R. M. Verma: Can Machines Tell Stories? A Comparative Study of Deep Neural Language Models and Metrics

FIGURE 29. Comparing all models for Sentence Connectedness by varying $k$ at constant $t$ where (a) $t = 0.5$. (b) $t = 0.75$. (c) $t = 1.0$.

significance study conducted for sentence connectedness measure is reported in Table 10.

From pilot studies for the different GPT2 variants (pre-trained and fine-tuned models), we observe that at varying values of $k$ at $t = 0.75$, the pre-trained model GM and GL show a change in coherence similar to human writing at the sentence level. Similarly for the fine-tuned models, w5-3M and w10-3M are the ones generating content similar to humans. The trend can be observed in the Figures 27a and 27b for the pre-trained models and Figures 28a and 28b for the fine-tuned models. Selecting the top models, we look at the generated textual content at different combinations of the sampling hyperparameters $k$ and $t$. The figures in this section show that the variation in the sentence connectedness values are closest to the human levels at the $k$ values greater than 100 and $t = 0.75$. The best performing models are the selected fine-tuned models (w5-3M and w10-3M) and GM.

Observations on sentence connectedness are below:

- Samples generated by the pre-trained model gpt2-medium and the fine-tuned wP_512_355M and wP_1024_355M at softmax temperature, $t = 0.75$ score are closest to the human content.
- The samples generated by the top models at the top-$k$ sampling values greater than 100, i.e. at 150, 250, 500, have variation in sentence connectedness closest to human written references.
- Changing $k$ at constant $t$ generates samples that do not fluctuate on coherence values. The gradient of the variation is more consistent for $t = 0.75$ and shows a more downward trend with $t = 1$.
- The connectedness scores for the XLNet based models show an upward trend with changing temperature at constant $k$ values while other models show a more downward trend.

XII. PART-OF-SPEECH USAGE

The distribution of parts-of-speech (POS) tags in textual content provides information like similarity in authorship, rarity of word usage and originality of POS tags. Here, we report the percentage of Noun and Verb tags that appear in the generated text with human stories acting as the baseline. The Spacy POS tagger\footnote{https://spacy.io/usage/linguistic-features} for Python was used for tagging purposes.

\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
$k$ & 0.25 & 0.5 & 0.75 & 1.0 \\
\hline
0 & & & & XH XL \hline
10 & & & & w10-3M, w5-3M \hline
50 & & & & w10-1M, w10-3M, w5-3M \hline
1000 & & & & XH \hline
\end{tabular}
\caption{(k, t) combinations of pre-trained models where Sentence Connectedness is statistically similar to human references.}
\end{table}

A. VERB USAGE DISTRIBUTION

We first discuss the statistical significance of the verb tag distribution in the generated content to the human references as shown in Table 11. The table shows that the samples from the pre-trained XL and OG and fine-tuned w10-1M and w5-3M models have verb distributions statistically similar to human references at different combinations of $k$ and $t$.
We perform the set of pilot studies to select the best performing pre-trained (PS-I) and fine-tuned (PS-II) GPT2 models.

The results of the pre-trained GPT models for PS-I are shown in Figures 31a and 31b for constant $t$ and $k$ respectively. The evaluation also reveals that openai-gpt and gpt2-large models have a more uniform plot with respect to the human baseline although the softmax temperature $t = 1.0$ is not suitable for the openai-gpt model. For the fine-tuned models in PS-II, the models w10-1M and w5-3M are the top performers with samples generated being consistently closer to the scores of human references as seen from Figures 32a and 32b. The best $k$ values were 150, 250 and
500 at \( t = 1.0 \), while temperature values are 0.75 and 1.0. For PS-II, we see that that the usage has a decreasing trend with an increasing value of \( k \) sampling parameter as shown by Figure 32a. The trend is reversed in Figure 32b as at \( k = 50 \), the increase in the temperature sees an increased verb frequency.

The analysis using the best models from pilot studies and the additional XLNet and Transformer-XL pre-trained models, are shown in Figures 33a, 33b and 33c for varying \( k \) at constant values of \( t \). The verb usage distribution for the chosen GPT2-based fine-tuned models resemble the human baseline at \( t = 0.75 \). The XB and XL models also generate samples that have verb tag distributions closer to human baseline. The results of the experiments by varying \( t \) at constant values of \( k \) are shown in Figures 34a, 34b, 34c, and 34e.

Our observations on Verb usage are:

- We see a decreasing trend in the occurrence of Verbs with an increasing \( k \) value at \( t = 1.0 \) for all the models. At \( t = 1.0 \) and \( k = 150 \), the models G2, GM and the fine-tuned w10-1M and w5-3M generate verbs at a rate similar to human writing references. The rate of verb usage is consistent with the changing values of \( k \) at \( t = 0.75 \) for all the models apart from TX (Figure 33a).
- Varying \( t \) at a constant value of \( k \) shows that the pre-trained models XL and fine-tuned w10-1M generate text with verb distributions similar to human references at \( t = \{0.5, 0.75\} \).
- The XLNet based models and the fine-tuned GPT2 models perform similar to human references on this metric. However, the statistical significance results and empirical evaluation reveal that verb usage in generated content is similar to human references. Therefore, this metric cannot differentiate between human and machine-generated writing.

**B. NOUN USAGE DISTRIBUTION**

Now we look at noun tag frequency in generated content as compared with the human references. As seen above, metric values change with sampling parameter values. Therefore,
TABLE 11. \((k, t)\) combinations of pre-trained and fine-tuned models where metric is statistically insignificant from human references.

(a) Verb Tag Distribution.

| \(k\) | \(t\) | \(0.25\) | \(0.5\) | \(0.75\) | \(1.0\) |
|---|---|---|---|---|---|
| \(0\) | OG, GL | OG, XL | XL, w5-3M, w10-1M | XL, w10-3M |
| \(10\) | XL, XL | OG, XL | XL, w5-1M, w10-3M | XL, w10-1M |
| \(50\) | XL | XL, w5-1M, w10-3M | XL, w10-1M | XL |
| \(250\) | OG, XL | OG, w5-1M | w10-1M | GL, XL, w5-1M |
| \(1000\) | GL, XL | OG, XL, w10-3M | XL, w10-3M | |

(b) Noun Tag Distribution.

| \(k\) | \(t\) | \(0.25\) | \(0.5\) | \(0.75\) | \(1.0\) |
|---|---|---|---|---|---|
| \(0\) | GM | GM | | |
| \(50\) | G2, GL | G2, GL, G2 | | |
| \(250\) | G2, GM, GL | G2, w5-1M, w5-3M | | |
| \(1000\) | G2, GM | w5-3M | | |

we consider different combinations of sampling hyperparameters \(k\) and \(t\).

Our observations on the statistical significance of the models for Noun tag distribution are in Table 11. It shows that the GPT-based models largely generate samples that are similar in scores to the human reference scores at higher values of softmax temperatures \(t \in \{0.75, 1.0\}\). The fine-tuned models generate better samples at \(t = 1.0\) at higher sampling value of \(k\) at 250 and 1000. We now study how these significance results compare with model performance at different temperatures and top-\(k\) sampling combinations.

For PS-I, we see how the frequency of the nouns generated by the pre-trained GPT2 models varies with change in sampling parameters \(- t\) and \(k\) as shown in Figures 35a and 35b respectively. The variation of the same in the samples generated by the fine-tuned models are shown in Figures 36a and 36b for varying \(k\) and \(t\) respectively.

A summary of the results comparing across samples generated across a comprehensive set of models is shown in the following experiments. Figures 37c, 37a and 37b show the changes in the metric with varying \(k\) at different values of constant \(t\). The changes with varying \(t\) at constant values of \(k\) are shown in Figures 38a, 38b, 38c, 38d and 38a.

Key takeaways on Noun tag distributions are:
- Samples generated by pre-trained G2 and GM have Noun tag distributions similar to human references at varying \(k\) values with constant \(t\) values.
- Fine-tuned models generate more nouns with an increase in temperature \(t = 1.0\) and higher values of sampling \(k\). At lower \(t\) values, the fine-tuned models generate fewer nouns than found in human writing.
- The generated samples tend to have more nouns than human references – the trend in the noun usage is almost consistent with changing hyperparameter combinations thus providing little information on how to differentiate them from human writing. Hence, noun usage is one of the lower ranked metrics as seen above.

XIII. MODEL RANKING

In the above experiments, we observed how the metric scores of the generated content compare with those of the
human-authored content. The most striking inference is that no model is close to human scores for every metric. For example, we observe that the TTR scores, on the text generated by the pre-trained models XLNet-large and XLNet-base models with varying $t$ (Figure 18), change notably across different values of constant $k$. This difference in trends among the models can be seen across varying sampling hyperparameters as well as metrics.

Thus to study model performance and stability, we compute two sets of deviation-based metrics – (a) TotalSD: The total standard deviation from the mean performance of the models across all combinations of metrics and sampling...
parameters; and (b) **TotalDevGH**: The total absolute deviation of the mean performance score of the model from the human score across all combinations of metrics and sampling parameters. We rank the models based on the above two metrics in the following Table 12, starting with the model that has the lowest TotalDevGH score.

We also calculate the Pearson’s product-moment correlation coefficient\(^\text{25}\) (\(\rho\)) to study the agreement between the two metrics reported in this section. The \(\rho\) value is 0.93, showing

\[^{25}\text{https://revisionmaths.com/advanced-level-maths-revision/statistics/product-moment-correlation-coefficient}\]
FIGURE 37. Comparing all models for Noun usage by varying $k$ at constant $t$ where (a) $t = 0.75$. (b) $t = 0.75$. (c) $t = 1.0$.

TABLE 14. An example showing a prompt and a generated story using the fine-tuned wP_512_355M model at $k = 150$ and $t = 1.0$.

| Prompt | Thicker lagoon |
|--------|---------------|
| wP_512_355M $k = 150, t = 1.0$ | As I approached the wave, the sub-pearl glow of the night reflected the shape of each life as I arrived close to the water’s edge. My ship docked at the head and the ship’s holographic display began to turn blue as all traces of time faded away. From the inside of the helmet, I knew my mission was accomplished. I had just accomplished what I had set out to accomplish in search of a few kind of energy for humanity. Walking toward the crew I realized, oh, what had happened to us. I lost all meaning in the form of time. Everything started happening again, I experienced life again for the first time. I understand this all had to be done. |

TABLE 15. An example showing a prompt and a generated story using the fine-tuned wP_1024_355M model at $k = 250$ and $t = 0.75$.

| Prompt | How do you get into the writing zone? Generated |
|--------|-----------------------------------------------|
| wP_1024_355M $k = 250, t = 0.75$ | I’ve been writing since I was just a kid, but I’ve been doing that for about five years now. I have a good number of friends that I’ve met through my work, and a few that I’ve met through doing writing. Now, I’ve met, writing is just an outlet for me to vent my frustration. I spend most of my time writing down what I want to accomplish, whether it be a poem, a character, or whatever, and set my brain do the rest. My motivation is mainly because I have a great plot, and a great character, and I want to see those two together. That’s the motivation I need to keep going. |

a strong positive correlation between the two sets of ranking metrics. The table shows that the fine-tuned GPT2 models and the Transformer-XL models are the ones reporting lowest A. Das, R. M. Verma: Can Machines Tell Stories? A Comparative Study of Deep Neural Language Models and Metrics
deviation in the metrics. The XLNet models have a high value of deviation, that agrees with our inference from the previous experiments specially the model scores from the readability metrics - FRE and DCR in Section X-A.

XIV. CONCLUSION

In this paper, we apply the large pre-trained deep neural language models such as OpenAI’s GPT and GPT2, Google/CMU’s Transformer-XL and XLNet to open-ended story generation and test their generalizability. We also compare performance of the fine-tuned GPT2 models (GPT2-117M and GPT2-355M) on two BPE-token based subsets of the WritingPrompts dataset.

Using a variety of automated metrics that measure linguistic, syntactic and semantic quality of the generated stories, the pre-trained and fine-tuned models are evaluated by comparing with human-written stories. Moreover, we also analyze the metrics with two techniques – a LASSO-based regression model and inter-metric correlation. In the exploratory analysis of metric importance, we see that the bigram based overlap measure is the best performing followed by the standard deviation in sentence length ($L_{sd}$).
Interestingly, retraining the model on a subset of the domain specific data enhances model performance. In our study, a model performs the best on a metric when the scores of the generated stories on that metric are statistically similar to that of human references. However, sampling parameters play an important role in text generation – higher values of softmax temperature $t$ (>0.5) and mid-range values of $k$ (50 < $k$ < 500) are the best parameter combinations. The top models are the pre-trained OpenAI-GPT-110M, GPT2-medium and Transformer-XL models along with GPT2-medium and GPT2-small models trained on Writing-Prompts-1024 subset data.
APPENDIX
We provide some examples of the text extracted generated by the pre-trained and fine-tuned models at different combinations of sampling parameters $k$ and $t$.
These examples are provided in Tables 13, 14 and 15.

ACKNOWLEDGMENT
The authors thank the reviewers for their insightful comments, that helped us improve the paper.

REFERENCES
[1] A. Rusia (2019). XNLet Speaks. Comparison With GPT-2. Accessed: Oct. 20, 2019. [Online]. Available: https://medium.com/amanrusia/xnlet-speaks-comparison-to-gpt-2-ca1a4e9a39e
[2] S. Baki, R. Verma, A. Mukherjee, and O. Gnawali, “Scaling and effectiveness of email masquerade attacks: Exploiting natural language generation,” in Proc. ACM Asia Conf. Comput. Commun. Secur., Apr. 2017, pp. 469–482.
[3] S. Bangalore, O. Rambow, and S. Whittaker, “Evaluation metrics for generation,” in Proc. 1st Int. Conf. Natural Lang. Gener. (INLG), vol. 14, Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 1–8. [Online]. Available: https://www.aclweb.org/portal/
[4] S. Bansal and C. Aggarwal. (2019). TextStat. Accessed: Oct. 20, 2019. [Online]. Available: https://pypi.org/project/textstat/
[5] T. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805
[6] A. Fan, M. Lewis, and Y. Dauphin. “Hierarchical neural story generation,” 2018, arXiv:1805.04833. [Online]. Available: http://arxiv.org/abs/1805.04833
[7] W. Fedus, I. Goodfellow, and A. M. Dai. “MaskGAN: Better text generation via filling in the...” 2018, arXiv:1801.07736. [Online]. Available: http://arxiv.org/abs/1801.07736
[8] X. Feng, M. Liu, J. Liu, B. Qin, Y. Sun, and T. Liu, “Topic-to-essay generation with neural networks,” in Proc. 27th Int. Joint Conf. Artif. Intell., Jul. 2018, pp. 4078–4084.
[9] D. Gavrilov, P. Kalaidin, and V. Malakh, “Self-attentive model for headline generation,” in Proc. Eur. Conf. Inf. Retr. Cham, Switzerland: Springer, 2018, pp. 97–93.
[10] M. Ghazvininejad, X. Shi, Y. Choi, and K. Knight, “Generating topical poetry,” in Proc. Conf. Empirical Methods Natural Lang. Process., Nov. 2016, pp. 1183–1191.
[11] A. Giaretta and N. Dragoni, “Community targeted phishing: A middle ground between massive and spear phishing through natural language generation,” 2017, arXiv:1708.07342. [Online]. Available: http://arxiv.org/abs/1708.07342
[12] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep Learning, vol. 1. Cambridge, MA, USA: MIT Press, 2016.
[13] A. Graves, “Generating sequences with recurrent neural networks,” 2013, arXiv:1308.0850. [Online]. Available: http://arxiv.org/abs/1308.0850
[14] L. Han, A. L. Kashyap, T. Finin, J. Mayfield, and J. Weese, “UMB_CBIQUITY-CORE: Semantic textual similarity systems,” in Proc. 2nd Joint Conf. Lexical Comput. Semantics, Main Conf. Shared Task, Semantic Textual Similarity (SEM), vol. 1. Atlanta, GA, USA: Association for Computational Linguistics, Jun. 2013, pp. 44–52. [Online]. Available: https://www.aclweb.org/anthology/S13-1005
[15] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, “The curious case of neural text degeneration,” 2019, arXiv:1904.09751. [Online]. Available: http://arxiv.org/abs/1904.09751
[16] D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck, “Automatic detection of generated text is easiest when humans are fooled,” 2019, arXiv:1911.00650. [Online]. Available: http://arxiv.org/abs/1911.00650
[17] G. Jagfeld, S. Jenne, and N. Thang Vu, “Sequence-to-Sequence models for data-to-text natural language generation: Word- vs. character-based processing and output diversity,” 2018, arXiv:1810.04864. [Online]. Available: http://arxiv.org/abs/1810.04864
[18] S. Karimi, L. Moraes, A. Das, A. Shakerly, and R. Verma, “Citation-based retrieval and summarization using IR and machine learning,” Scientometrics, vol. 116, no. 2, pp. 1331–1366, Aug. 2018.
[19] C. Kiddon, L. Zettlemoyer, and Y. Choi, “Globally coherent text generation with neural checklist models,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2016, pp. 329–339.
[20] J. P. Kincaid, R. P. Fishburne, Jr., L. R. Rogers, and B. S. Chissom, “Derivation of new readability formulas (automated readability index, fog count and Flesch reading ease formula) for navy enlisted personnel,” Naval Technical Training Command Millington TN Research Branch, Tech. Rep., 1975. [Online]. Available: https://stars.library.ucf.edu/sislibrary/56/
[21] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky, “Deep reinforcement learning for dialogue generation,” 2016, arXiv:1606.01541. [Online]. Available: http://arxiv.org/abs/1606.01541
[22] C. Liu, S. He, K. Liu, and J. Zhao, “Curriculum learning for natural answer generation,” in Proc. 27th Int. Joint Conf. Artif. Intell., Jul. 2018, pp. 4223–4229.
[23] E. Loper and S. Bird, “NLTK: The natural language toolkit,” 2002, vol. 1. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://arxiv.org/abs/1706.01399
[24] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1532–1543.
[25] M. E. Peters, M. Neumann, M. Iyyer, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” 2018, arXiv:1802.05365. [Online]. Available: http://arxiv.org/abs/1802.05365
[26] R. D. Powers, W. A. Sumner, and B. E. Kearl, “A recall of four adult readability formulas,” J. Educ. Psychol., vol. 49, no. 2, p. 99, 1958.
[27] A. Rachkovskij, D. D. Politov, A. Bar, B. Ermoliev, J. Benavoli, and L. Wolf, “Language generation with recurrent generative adversarial networks without pre-training,” 2017, arXiv:1706.01399. [Online]. Available: http://arxiv.org/abs/1706.01399
[28] Q. Qian, M. Huang, H. Zhao, J. Xu, and X. Zhu, “Assigning Individuality/Profile to a chatting machine for coherent conversation generation,” in Proc. 27th Int. Joint Conf. Artif. Intell., Jul. 2018, pp. 4279–4285.
[29] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” OpenAI Blog, vol. 1, no. 8, 2019.
[30] E. Reiter and A. Belz, “An investigation into the validity of some metrics for automatically evaluating natural language generation systems,” Com- put. Linguistics, vol. 35, no. 4, pp. 529–558, Dec. 2009.
[31] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “Hierarchical latent vector model for learning long-term structure in music,” 2018, arXiv:1803.05428. [Online]. Available: http://arxiv.org/abs/1803.05428
[32] M. Roemmele, A. S. Gordon, and R. Swanson, “Evaluating story generation systems using automated linguistic analyses,” in Proc. Workshop Mach. Learn. Creativity (SIGKDD), 2017, pp. 13–17.
[33] D. Scott and D. Hardcastle, “Can we evaluate the quality of generated text?” in Proc. LREC, May 2008, pp. 1–8.
[34] A. See, A. Pappu, R. Saxena, A. Yerukola, and C. D. Manning, “Do massively pretrained language models make better storytellers?” 2019, arXiv:1909.10705. [Online]. Available: http://arxiv.org/abs/1909.10705
R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Artzi, “BERTScore: Evaluating text generation with BERT,” 2019, arXiv:1904.09675. [Online]. Available: http://arxiv.org/abs/1904.09675

AVISHA DAS received the B.Tech. degree in electronics and communication engineering from the West Bengal University of Technology, West Bengal, India. She is currently pursuing the Ph.D. degree with the Department of Computer Science, University of Houston. Her research interests include natural language understanding and generation, cybersecurity, and information retrieval.

RAKESH M. VERMA (Member, IEEE) is currently a Professor of computer science with the University of Houston (UH), where he is leading a research group that applies reasoning and data science to cybersecurity challenges. He is the coauthor of Cybersecurity Analytics (CRC Press, 2019), which discusses key data analytics techniques and their adaptations for cybersecurity challenges. He has co-organized the 1st Anti-phishing Shared Task, in 2018, with proceedings in the CEUR workshop series; the training dataset is publicly available for academic research upon request. Since 2015, he has been co-organizing and editing the proceedings of the ACM International Workshop on Security and Privacy Analytics. He is an Editor of Frontiers of Big Data in the Cybersecurity Area, an ACM Distinguished Speaker, from 2011 to 2018, and the winner of two Best Paper Awards. He received the 2013 Lifetime Mentoring Award from the University of Houston. He is a Fulbright Senior Specialist in computer science, a Panel Chair of ACM CODASPY 2020, and a Technical Program Committee Co-chair of ACM CODASPY 2021.