Gen AI Text Segmentation

Jaideep Nuvvula nuvvula@usc.edu
Jae Park jaewoop@usc.edu
Swaraj Vatsa vatsa@usc.edu
Huzefa Ali huzefaal@usc.edu
Ravi Basireddy rbasired@usc.edu

Abstract

With recent advancements in complex machine learning algorithms, it has become increasingly effortless for models to generate text that closely mimics human writing. However, even the most advanced language models can sometimes produce text that contains subtle inconsistencies that keen readers can discern. Our approach continues a line of work that aims to precisely determine the boundary where a human-written prompt concludes and a machine-generated continuation begins. These findings can impact various domains, including journalism, law, finance, and academia, where work integrity can be tampered with by machine text. We have released our code on Github[1].

1 Introduction

The prevalence of hybrid documents containing human-written and machine-generated text has increased significantly in recent years. Identifying and distinguishing between human and machine-generated content within these documents has become crucial in various fields. However, the traditional manual method of detecting hybrid human-machine text boundaries is time-consuming and requires human experts to examine and manually annotate the text meticulously.

With the advent of natural language processing (NLP), the detection of hybrid human-machine text boundaries can be automated, providing a more efficient and effective way of identifying and distinguishing human and machine-generated text. This research paper presents a comprehensive study on the automatic detection of hybrid human-machine text boundaries.

This research project seeks to assess the effectiveness of various methods in detecting the boundary between human-written and machine-generated text in hybrid documents. This study particularly utilizes the attention mechanism (Bahdanau et al., 2014) in conjunction with sentence embeddings generated by the DistilRoBERTa (Sanh et al., 2019) language model. Specifically, the attention mechanism is applied to the sentence embeddings to produce weighted embeddings, which are subsequently fed into various classifiers to detect the boundaries between human-written and machine-generated text.

Our investigation also aims to identify limitations in the existing approaches and areas for potential enhancement. One notable drawback of the current approaches is their oversight of important details of the context surrounding the sentence in a given passage. We hypothesize that the surrounding structure of a sentence can provide valuable insights into the origins of a passage. The attention mechanism is part of a neural architecture that enables the dynamic highlighting of relevant features of the input data. In NLP, the input data is typically a sequence of textual elements, and the idea behind attention is to compute a weight distribution on the input sequence, assigning higher values to more relevant elements.

Our results confirm that the weighted embeddings as input tend to capture the hidden *patterns* more effectively and achieve better accuracy in identifying the boundary compared to the work by (Cutler et al., 2021). Our findings' implications can significantly impact various fields that utilize hybrid human-machine documents, such as journalism, legal documentation, and academic research. The automated detection of human-machine text boundaries can provide a more efficient and effective way of processing large volumes of text and improve the accuracy of the analysis.

2 Related Work

The identification of machine-generated text has been an active area of research in recent years, focusing on distinguishing between human and

[1] https://github.com/jaeparkim/ShallowMinds

machine-generated text. Additionally, a new line of research concentrates on identifying boundaries in hybrid documents where machine-generated text and human-generated text merge. This section will provide an overview of the methods used for detecting machine-generated text, including feature-based and neural language-based methods. Later, we will discuss the approach to identifying boundaries in hybrid documents.

Feature-based approaches are rule-based methods for detecting machine-generated text. These methods use various linguistic features, including word frequency, fluency(Holtzman et al., 2019; See et al., 2019), and other features such as the Gunning-Fog Index, Zipf's law (Zipf, 1949), TF-IDF (Radford et al., 2019a), and POS tag distributions, to distinguish between human and machine-generated texts. High-level characteristics such as punctuation and sentence length can also be utilized. However, these methods have limitations, such as being vulnerable to evasion, lacking context, dependency on the source language, and limited scalability.

The neural language model approach is a more effective way of detecting machine-generated text. This approach involves incorporating attributes derived from Transformer language models. Two well-known neural language model approaches exist for generated text detection: zero-shot and fine-tuning approaches.

The zero-shot approach uses generative models such as GPT-2 (Radford et al., 2019b) or Grover (Solaiman, 2019; Zellers, 2019) without fine-tuning. This approach involves appending a classification token ([CLS]) to the end of the input sequence, and the token embedding represents a feature vector on the entire input sequence. These feature vectors can be used to train a linear layer of neurons to classify whether a machine or a human produced the input sequence. However, this approach is limited because it relies on autoregressive models like GPT-2 (Radford et al., 2019b) or Grover, which are unidirectional and less effective at detecting machine-generated text from other bidirectional models, such as BERT.

The fine-tuning approach involves fine-tuning the language model on a binary classification task of distinguishing between human and machine-generated text. It is typically done by adding a classification layer to the pre-trained language model and training it on a small labeled human and machine-generated text dataset. One of the constraints of the fine-tuning approach is that it is limited to specific model architectures and requires labeled data. Additionally, the quality of the dataset used for fine-tuning can significantly impact the model's performance.

In identifying boundaries in hybrid documents, sliding window approaches segment the text into consecutive sentence pairs. Each pair is then classified as either human-written or machine-generated. After identifying the breakpoints, the boundaries are refined by evaluating the optimality of the identified points, in which human annotators often get involved. Sentence fragments or incomplete sentences are discarded, and adjacent boundaries separated by a small number of sentences or words are merged.

The current state-of-the-art model(Cutler et al., 2021) for boundary detection utilizes Sroberta (Liu et al., 2019) to generate sentence-level embeddings. These embeddings are concatenated to create a passage embedding, which is then fed into a logistic regression classifier. This approach has been effective in detecting boundaries in hybrid documents.

In conclusion, researchers have proposed different approaches to detect machine-generated text and identify boundaries in hybrid documents. The feature-based and neural language-based methods discussed in this section have demonstrated their effectiveness in detecting machine-generated text. At the same time, the sliding window approach has been helpful in identifying boundaries in hybrid documents.

3 Data

Our project utilizes the "Real or Fake Text" (RoFT) dataset (Dugan et al., 2020). The dataset presents human-written text followed by machine-generated text to test the ability to identify the boundary between them. In other words, the task was to determine the last line the human wrote within a given passage that maintains context.

The RoFT dataset consisted of 42,165 data points, from which we removed duplicates and normalized the true boundary values to create a final dataset of close to 9139 data points.

The human prompts in the RoFT dataset were sourced from four distinct genres of text: news articles, presidential speeches, fictional stories, and recipes. The news articles were sourced from the New York Times Annotated Corpus, a diverse col-

lection of over a million articles from 1987 to 2007. The presidential speech corpus was developed by Brown in 2016 and included 963 speeches dating back to 1789. The RoFT dataset also drew on fictional stories from the Reddit platform and the Recipe1M+ dataset, which contains over a million recipes from various online sources. The distribution of the sources is shown in Figure 1.

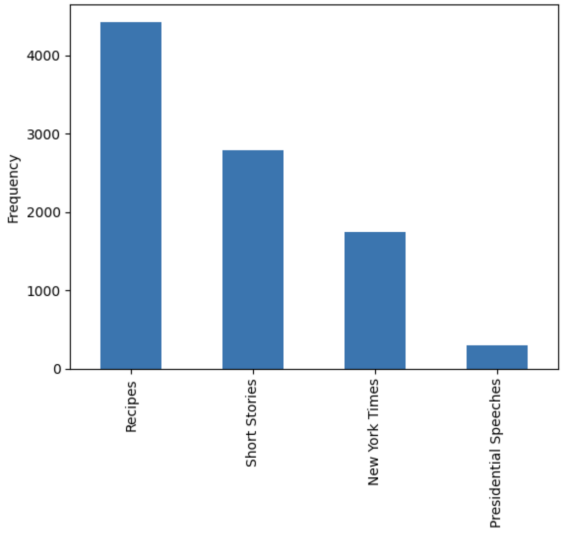

Figure 1: The distribution of genres for human-written text

To generate continuations in RoFT, the authors used various Language Models (LLMs), such as GPT-2 (Radford et al., 2019b) base, GPT-XL (Radford et al., 2018), and CTRL model(Keskar et al., 2019), among others. They employed the nucleus sampling parameter of p, which ranges from 0 to 1 (Holtzman et al., 2020), and set the repetition penalty to 1.2 (Keskar et al., 2019). The distribution of this is shown in Figure 2.

Using RoFT, which includes multiple text genres and different LLMs for continuations, helps the model identify the boundaries of human-written text across various writing styles and contexts. This approach provides a more comprehensive analysis of text data and enhances the model's performance in distinguishing between real and fake text.

4 Methods

4.1 Random Sampling & Majority Class

To establish a baseline for our project, we used the same baselines described in (**?**): a random and

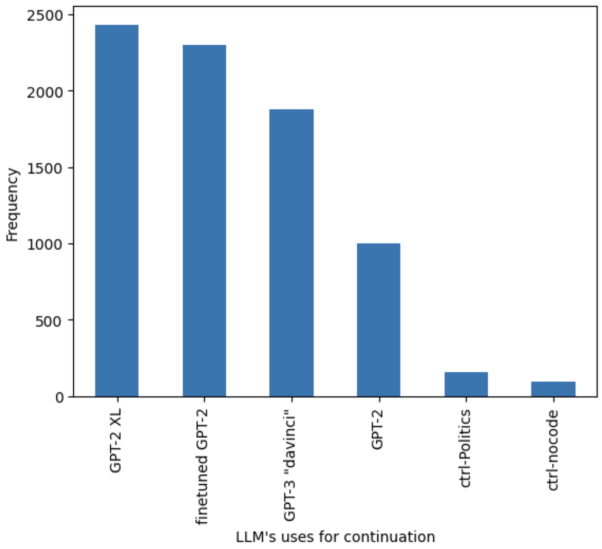

Figure 2: The distribution of LLMs used for continuing human prompts

majority-class model. The purpose of the majority class model is to detect any potential class imbalances in the RoFT dataset, which could affect the accuracy of our boundary detection model. On the other hand, the random model serves as a reference for the worst possible performance of a classifier.

The majority class model is a simple classifier that assigns all instances in the dataset to the most frequently occurring class. By comparing the performance of our boundary detection model to the majority class model, we can determine if any significant class imbalances need to be addressed.

The random model is a classifier that randomly assigns a class to each instance in the dataset. This model serves as a benchmark for the performance of our boundary detection model. By comparing the performance of our model to the random model, we can establish a baseline for an acceptable level of minimal-achievable accuracy.

4.2 Semantic Textual Similarity

Our first method involves the generation of sequential embeddings using SRoBERTa (Liu et al., 2019). To generate sequential embeddings, we take each sentence in the passage and generate an embedding of it along with all its preceding sentences. For instance, the embedding for sentence three would be generated using the text of sentences 1, 2, and 3, allowing us to capture the semantic information of all preceding sentences up to that point.

Once we have generated the sequential embeddings, we calculate their pairwise cosine similarity.

By doing this, we are comparing the overall semantic similarity of a sentence with the cumulative semantic information of all the preceding sentences in the passage. This approach helps capture longer-range semantic relationships between sentences, such as the overall coherence and flow of the passage.

The resulting embeddings are fed into a Random Forest classifier to find the true breakpoint between human-written and machine-generated text.

For the extraction of SRoBERTa (Liu et al., 2019)embeddings, we used the HuggingFace transformers package. Finally, for the Random Forest classifier, we used the scikit-learn library.

4.3 Sentence-level perplexity

In this method, we use Sentence-level perplexity scores to distinguish between human-written and machine-generated text. Perplexity is a standard measure used to evaluate the quality of a language model by assessing its ability to predict the likelihood of a sequence. It is calculated by taking a given sequence's exponential average negative log-likelihood.

Our study computed sentence-level perplexity scores using the HuggingFace transformers package. These scores were then utilized as features in a Random Forest classifier, to accurately identify the breakpoint between the two types of text.

4.4 Attention on Distilroberta embeddings

To our knowledge, for the current SOTA model in boundary detection (Cutler et al., 2021), a fixed-length sentence embedding of size 768 is generated using the DistilRoBERTa model. This embedding is generated for each sentence in the dataset, and all the embeddings are concatenated together into a 7,680-dimensional passage embedding. This passage embedding is then fed into a logistic regression classifier to detect boundaries in both human-written and machine-generated text.

However, there is a significant drawback to this approach. Sometimes, the sentences in the dataset need to be longer to provide enough information to generate meaningful embeddings. Additionally, to capture the context of a sentence, it is necessary to consider the sentences that come before and after it. Therefore, to address these issues, we propose an attention model.

In our attention model, we use the fixed-length sentence embeddings generated by the DistilRoBERTa model and apply weights to them based on their importance in the context of the passage. It allows us to capture the semantic meaning of each sentence and its relationship to the surrounding sentences. The weighted embeddings are then used as input to a neural network for classification.

5 Experiment Results & Discussion

We split our dataset into three subsets: training, validation, and test sets, using an 80:10:10 split. The purpose of splitting the dataset was to train and tune the models on the training and validation sets, respectively, and then to evaluate the models on the test set, which had yet to be used during the training or tuning process.

To measure the accuracy of our models, we chose to use the standard 0/1 error classification accuracy metric, which only considers the number of exactly-correct predictions and disregards slightly incorrect ones. This accuracy metric aligns with using cross-entropy loss as the loss function. Therefore, we aimed to give equal weight to slightly incorrect and significantly incorrect predictions.

To ensure that our results were reproducible, we gave all of our models the same random seed (42069) and reported the accuracy of each model on the test set. Once we selected the models and tuned the hyperparameters using the validation set, we conducted all our experiments on the test set. The results for each of the methods are shown in Table 1.

Model	**Test Accuracy**
Random Sampling	10.1%
Majority Class	14.8%
Semantic Textual Similarity	21.7%
Sentence-level Perplexity	21.3%
Attention w/ DistilRoBERTa	**30.4%**

Table 1: Accuracy of different models

5.1 Random Sampling & Majority Class

As hypothesized, the random baseline model achieved a near-uniform accuracy of approximately 10 percent across all data subsets and on the full dataset. However, a more informative baseline model was the majority class baseline, bringing to light the class imbalance issue in the dataset. Specifically, a significant proportion of the dataset, comprising 15 percent of the samples, was labeled

as class 9, indicating that the true boundary breakpoint is at the end of the passage and that the entire passage was human-written.

The class imbalance underscores the importance of appropriate sampling and data preprocessing techniques to mitigate bias and improve model performance.

5.2 Semantic Textual Similarity

In Table 1, we present the accuracy results for our methodology in measuring semantic textual similarity (STS). Our test set yielded an accuracy of 21.7 percent, indicating the effectiveness of our approach in capturing semantic relationships between sentences.

To better understand the performance of our model, we selected a representative data point from the test set for which our model accurately predicted the true boundary. In Figure 3, we plot the cosine similarity value for each sequence, with the true boundary for the example being at position 5. We observed a noticeable curvature at the breakpoint, indicating a significant change in the semantic content of the text from human-written to machine-generated.

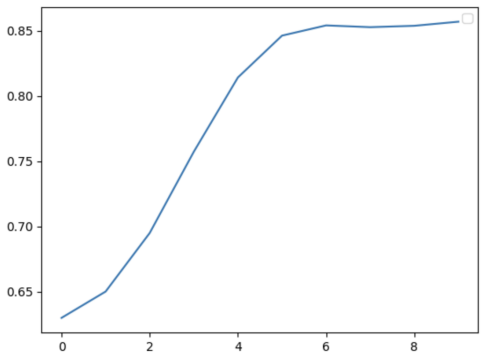

Figure 3: Sample plot of STS values at each breakpoint, true breakpoint being 5

We also investigated a data point for which our model could not accurately predict the true breakpoint, as shown in Figure 4. For this example, the true breakpoint was at position 7, but our model predicted it to be at position 6. In contrast to the previous example, we observed a much smaller curvature at the predicted breakpoint, suggesting that our model struggled to capture the change in the semantic content of the text accurately.

5.3 Sentence-level perplexity

Our findings show that the success of the perplexity baseline approach is heavily reliant on the specific

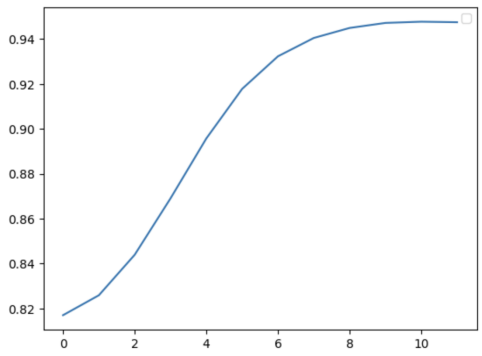

Figure 4: Sample plot of STS values with the true breakpoint at 7

top-p sampling technique used for generating text. In our experiments with GPT2-XL (Radford et al., 2019b), when p=0.0, the perplexity approach outperformed the majority baseline by approximately three folds. However, as the value of p increased, the perplexity method's performance declined. It was due to the generated text becoming less predictable and more closely resembling the actual distribution of sentences written by humans. When p=1.0, the perplexity of machine-generated text surpassed that of human-generated text, resulting in a higher perplexity for machine-generated sentences. This, in turn, led to our model being unable to accurately predict the point where perplexity increases, resulting in a decline in detector performance worse than random guessing. Table 2 shows the Test accuracy of the model on different values of p.

Top-P value	Test Accuracy
0.0	39.4%
0.4	23.2%
1.0	7.6%

Table 2: Accuracy of the perplexity model for different Top-P Values

5.4 Attention on Distilroberta embeddings

Our experimentation with different methods revealed that using attention on Distilroberta embeddings resulted in the highest accuracy on the entire RoFT dataset, achieving a score of approximately 30.4 percent. While we are still analyzing the reason for the higher accuracy compared to other methods, our current assumption is that our model captured the semantic meaning of each sentence and

its context within the passage. This capability allowed our model to accurately detect boundaries in various text types and styles, which is particularly important for natural language processing tasks. Figure 3 displays the validation accuracy for the first 100 epochs of our model.

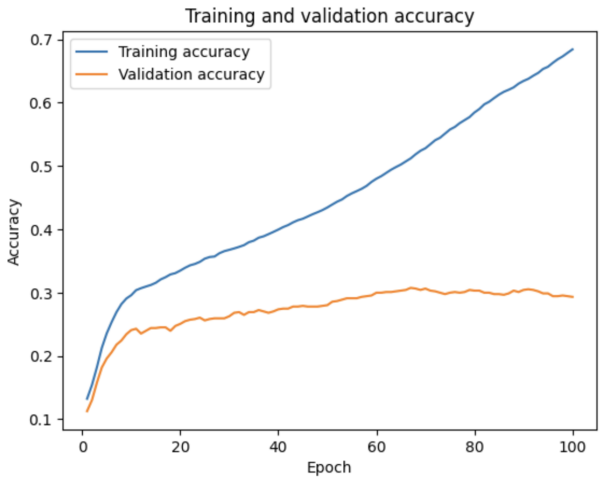

Figure 5: Accuracy of the Attention model on Train and Valid over 100 epochs

6 Future Works

In this paper, we demonstrate the viability of the boundary detection task as a framework for learning and detecting the differences between human text and machine-generated text. This framework is more challenging and robust than the passage-level binary classification task. That is, given a passage of arbitrary length, determine whether the entire text block is machine-generated or human-written. A study by (Solaiman et al., 2019) reported that fine-tuning a RoBERTa model (Liu et al., 2019) as a classifier for texts by humans and GPT-2 (Radford et al., 2019b) achieved over 99% accuracy on its binary classification task. Although this result is encouraging, it is worth noting that the binary detection task is significantly easier than the boundary detection task. The "Real or Fake Text" (RoFT) dataset (Dugan et al., 2020) is a well-known dataset for benchmarking the boundary detection task. The first several sentences of each data point are human-written text collected and refined from various sources, such as News, recipes, and Reddit posts. Following this, the several subsequent sentences are a machine-generated continuation of the preceding passage. Finally, the dataset only consists of 7257 true labels, which were then further partitioned into train (80%), valid (10%), and test (10%) data. Future work will seek to advance the experiment by using data augmentation to improve the true boundary detection of our model. Figure 6 illustrates a proposed method for augmenting the human-written and machine-generated text without shuffling or word replacement. The sentences are sequentially appended for each human-written sentence until we reach the true boundary, which is after the fourth sentence in this example. Then, the process repeats from the fifth sentence, the first sentence of the machine-generated part of the passage, until the end. This method was proposed to preserve as much semantics and context as possible from the original texts. The results from applying this proposed method will verify whether the unsatisfactory performance of our model was partly due to relatively small training data. We suggest further improvements in future work by employing Transformer blocks using self-attention (Vaswani et al., 2017). The current implementation uses the attention mechanism, but we expect that self-attention will be better at retaining information and looking at the whole context of the sequences. In other words, self-attention might reveal the possibility of hidden-meta patterns in LM-generated text by attending to different parts of its input sequences rather than assessing the relationship across multiple sentences.

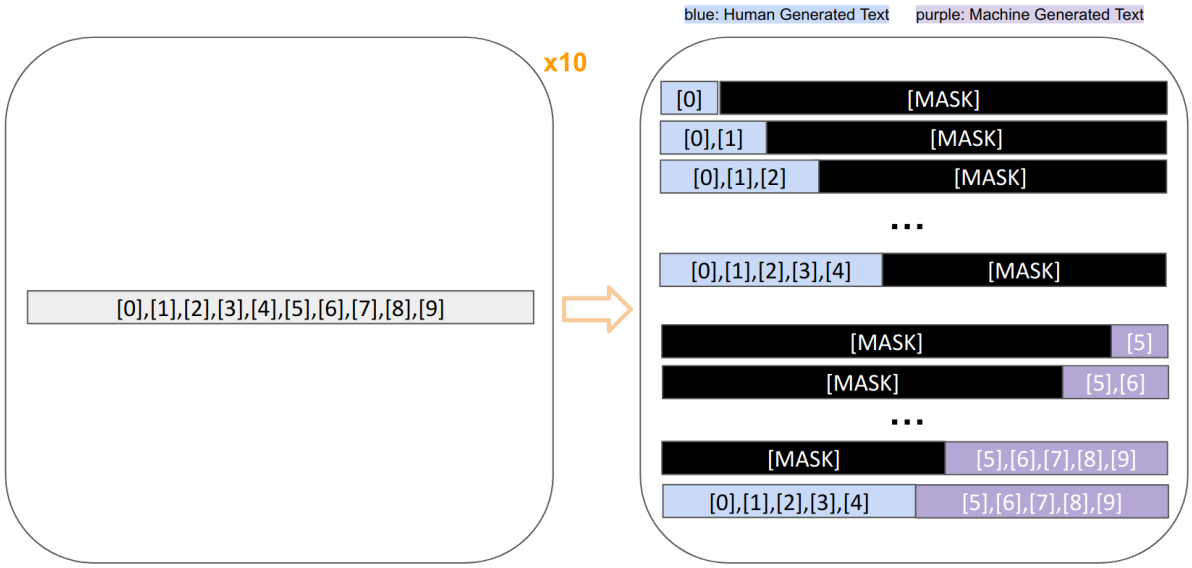

Figure 6: suggested data augmentation method for future work

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Joseph Cutler, Liam Dugan, Shreya Havaldar, and Adam Stein. 2021. Automatic detection of hybrid human-machine text boundaries. *arXiv preprint arXiv:2104.03888*.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. Roft: A tool for evaluating human detection of machine-generated text.

Ari Holtzman, Jan Buys, Li Du, Maggie Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration.

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

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. In *Proceedings of the 2018 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1–15. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019a. Language models are unsupervised multitask learners. Technical Report 1901.09913, OpenAI, San Francisco, CA, USA.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019b. Language models are unsupervised multitask learners. *OpenAI Blog*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4364–4374.

Abigail See, Aniruddh Pappu, Rajiv Saxena, Ashwin Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? *arXiv e-prints*.

Intisar Solaiman. 2019. Release strategies and the social impacts of language models. *arXiv e-prints*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Tom Henighan, Rewon Child, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Rowan Zellers. 2019. Defending against neural fake news.

G. K. Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley Press.