GPT-3 as a Text Summarizer

Tae Hyoung-Jo and Josh Ip

Abstract

There has been a recent trend towards pre-training language representations in Natural Language Processing (NLP) in an effort to generalize systems. Even still, there remains the need for task-specific datasets of tens of thousands to hundreds of thousands of examples along with a computationally expensive process of fine-tuning for state-of-the-art performance. In contrast, humans perform new language tasks well from only a few examples or with nothing but simple instruction – defined respectively as "few-shot" and "zero-shot" learning, still a difficult task for most NLP models and a focus of much recent literature. GPT-3 is a language model built by OpenAI and represents the cutting age of the new few-shot paradigm. While OpenAI highlights GPT-3's performance in tasks like text generation and simple arithmetic, one central NLP task not measured is GPT-3's ability to summarize text. This paper hopes to extend the benchmarking done on GPT-3 and examine its performance on various summarization tasks. The paper benchmarks OpenAI's summarization performance on the Gigaword and Daily Mail/CNN datasets, using both the ROUGE metric and a human evaluation, designed to address the inability of ROUGE to measure characteristics like fluency and grammatical correctness. We measure GPT-3's performance between abstractive and extractive tasks, between different methods of "prompt design", and between the four available GPT engines

1. Introduction

Our research attempts to resolve two research problems: GPT-3's lack of text summarization benchmarks and the inflexibility of SOTA summarization models.

In order to introduce and describe the approaches of our problem, we must also introduce GPT-3, extractive and abstractive summarization, ROUGE, and its shortcomings.

1.1. GPT-3

In "Language Models are Few-Shot Learners," Brown, Mann, Ryder, Subbiah at. introduce GPT-3, a Transformer based language model trained with 175 billion parameters, a whole order of magnitude more than its closest predecessor. As input, GPT-3 takes a tokenized string, with which the researchers design "prompts", moving away from the paradigm of fine-tuned models and instead passing in as input both data examples and directives. The researchers show how this "few-shot" strategy along with GPT-3's massive scale leads to surprisingly good performance across a number of highly different tasks that rival fine-tuned state of the art models. These comparisons were done on a few sets of tasks. The model demonstrated GPT-3's ability to produce human-passable news articles, do arbitrary math, and few-shot learning tasks.

1.2. Summarization

In short, summarization is the task of taking a section of text and creating a fluent and concise transformation of that text that includes key information and preserves meaning. [1] There are two key subtypes of summarization: extractive and abstractive. Extractive takes the section of texts and produces a summary formed out of the keywords from the original text. Abstractive focuses on using NLP techniques to reproduce the important material in a novel way with a focus on naturality instead of simply refactoring the original text. [2] While extractive summarization has been heavily researched, abstractive summarization has now become the focus.

1.3. ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) was first introduced by Chin-Yew Lin in "ROUGE: A Package for Automatic Evaluation of Summaries." in which the author discusses the four ROUGE measures (ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S) and discusses the measures' effectiveness as an automatic metric for summarization. [3] ROUGE relies on datasets that have a set of text paired with a reference summary (usually produced by humans). ROUGE then benchmarks a systems summary (or machine produced summary) against the reference summary and focuses on two quantitative measures: precision and recall. Recall is defined as the ratio of the number of overlapping words between the reference and system summary divided by the total number of words in the reference summary. However, this does not account for whether the systems summary could just be extremely verbose, so the study also uses precision: the number of overlapping words divided by the total numbers of words in the system

summary. The difference between the ROUGE-N metrics is that the metric looks at precision and recall for n-grams. ROUGE is used for both datasets designed for abstractive and datasets designed for extractive evaluation. This paper will use the GigaWord dataset with ROUGE and human evaluation to benchmark GPT-3's performance in extractive summary We will use the CNN / Daily Mail dataset for evaluating abstractive summary using ROUGE. [3] While these datasets will assess initial benchmarkings of GPT-3's ability to summarize – one often criticism of ROUGE is its specific focus on pure lexical comparison. We contend that summarization should also be tested on its naturalness and ability to pass as human written. In order to do that, we must introduce a novel qualitative test.

2. Background and Related Work

To better contextualize the research in this paper, further background can be given on the abstract and summarization methods, including an example start of the art method of both. Further, the GPT-3 architecture will be explained along with some additional related work on ROUGE, qualitative summarization evaluation, and the specifics of OpenAI's different engines.

2.1. Abstractive Summarization Method State of the Art

We explore some examples of the state of the art. In Xiao et. al's paper "ERNIE-GEM: An Enhanced Multi-flow Pre-training and Fine-tuning Framework for Natural Language Generation", the authors describe effects exposure bias has to downstream tasks. [4] In order to make the summarization closer to human-writing, the model trains text span-by-span as opposed to word-by-word to generation which helps the model improve by context. The performance of this model when in abstractive summarization was benchmarked as state of the art against Gigaword and CNN/DailyMail even when trained with a much smaller amount of pre-training data. GPT-3 also being found to be successful at contextual learning suggests similar success for the model.

2.2. Extractive Summarization Method State of the Art

We can briefly examine state of the art for extractive summarization. In "Better Fine-Tuning by Reducing Representational Collapse" by Aghajanyan et. al, the research focused on a trust region method that generally examines their performance and shows improvements in text summarization when benchmarked against the Gigaword dataset. [5] We will compare GPT-3's performance against these state of the art.

2.3. Human testing

Whilst the ROUGE method will allow one to compare the results of each experiment, ultimately the scoring system is fairly brittle and tough to measure abstractive data. Therefore, one can also plan on instigating a human evaluation test which will gain qualitative and quantitative data about GPT-3's performance.

Using the framework set forward by "Controlling the Amount of Verbatim Copy in Abstractive Summarization," we take the results of our few-shot davinci engine training on the GigaWord dataset and set up a survey. [6] For each summarization, we asked humans to rate the result's informativeness, grammaticality, truthfulness. We establish a control by creating our own human summarizations and having participants score and also guess which summarization is written by a human. We publish our own results and also compare them with the results given in the Song et. al paper.

Combined with our quantitative results from the ROUGE score, the human evaluation will allow us to further evaluate the performance of GPT-3 against the state of art.

2.4. GPT-3 Method

In order to further understand GPT-3's model and understand its state of the art few-shot learning tasks, we must first talk about the approach of the OpenAI paper in which it was introduced.

In principle, GPT-3 is a scaled up version of GPT-2 in terms of number of parameters trained. Using what amounts to a transformer architecture, 2048 words (or tokens) are inputted in and 2048 guesses are outputted. 2048 words is fixed and the sequence is repeated for longer tasks. At a high level, the architecture is a series of matrix multiplications with some algebraic steps.

The first step is to use Byte Pair Encoding (for efficiency) to one-hot encode the words into a vector of numbers to GPT's vocabulary of 50,527 words. Next, the vector is embedded into 12.288 dimensions so the size of the matrix is reduced. After the embedding, the tokens are further positionally encoded to be prepared for the multi-head attention where the process is repeated 96 times.

At a high level, the next step, the attention sequence, is helping to predict which input tokens to focus on for every output in the sequence. The model first learns three weight matrices to transform the sequence embedding into 3 separate matrices. The first two matrices are multiplied together (the "queries" and "keys") to make a 3x3 matrix that demonstrates the importance of the token (and interestingly is the only cross-word operation in the sequence). This process is repeated 96 times and then goes through a standard feed forward with one hidden layer to go through the process of adding the input to the output.

Finally, the result is normalized (Add & Norm) and is decoded which is basically the reverse of the original encoder and soft-maxed to find the resulting probabilities of each word in the sequence!

2.5. OpenAI's Different Engines

OpenAI's API provides four engines to use. Whilst DaVinci is the most generally capable engine, all four engines have their cost tradeoffs in terms of speed, cost, and performance. DaVinci costs $0.06 per thousand tokens, Curie costs $0.006 per thousand tokens, Babbage costs $0.0012 per thousand tokens, and Ada costs $0.0008 per thousand tokens. [7]

2.6.1 The DaVinci engine is known to be the best performer with the least amount of instruction. The engine specialties include "complex intent, cause and effect, summarization for audience." While most powerful, the engine is also the most slow and costs the most in terms of the API. Importantly, DaVinci is great at understanding cause and effect – but it's powerfulness in summarization will be tested in these experiments.

2.6.2 Curie, while powerful, is also very, very fast. While not as strong as Davinci in terms of processing complicated text – Curie can act as a great Q&A chatbot.

2.6.3 Ada is the fastest, but most basic engine that can only handle the least nuanced kind of tasks. Ada does improve with context, but its performance is inferior to Curie and DaVinci

2.6.4 Babbage is great at straightforward, simple classification and is also great at semantic search classification.

3. Approach

For the Gigaword dataset, we benchmark on the following variables: the choice of engine, the number of examples included in the input (that is, whether we adopt a zero, one, or few-shot approach) and the choice of "strategy" (different prompt designs). Additionally, we test different cross sections of these variables, for example, comparing the performance of the zero, one, and few shot approaches for each of the four engines. For each of these tests, we measure the ROUGE-1, ROUGE-2 and ROUGE-L scores. We also test whether we can take advantage of prompt design to ask GPT for summaries with specific parameters, testing whether GPT correctly responds to requests for one versus two sentence summaries, and whether it responds to requests for five versus ten word summaries.

We follow this up with a more subjective human evaluation.

3.1 Setup and Testing Environment

Our tests were run in Google Colab and through Josh's OpenAI API key. Dataset retrieval and ROUGE scoring used implementations and scaffolding tools provided by open source project Huggingface.

3.1.1 Helper functions

We have as scaffolds three core functions, the zero shot, one shot, and few shot approaches, each taking in a choice of strategy or custom strategy, the desired number of inputs to test, the number of tokens to request the API (where each token represents roughly 4 characters). As the default, we set 12 tokens as the number of tokens the API should return as 12 tokens was about the maximum number that the test data had as a summary.

For the one-shot and few-shot approaches, we take in training data and randomly select the necessary number of examples for each call of the API.

3.1.2 ROUGE scoring

We record specifically the ROUGE-1, ROUGE-2, and ROUGE-L scores for each test. For each, huggingface's implementation of ROUGE uses sampling to get confidence intervals (low, mid, high) for each of ROUGE-1, ROUGE-2, and ROUGE-3, and stores the precision, recall, and f-measure for each of these intervals

3.2 Approach: Gigaword

3.2.1 Testing choice of engine

Because the four OpenAI engines each have their own strengths and weaknesses in terms of power, speed, and cost we benchmark the engines against each other to find the best relative summarizer. To do this, we ran the same standard tests across all four engines and graphed them comparatively to understand their relative performance.

3.2.2 Testing Zero, One, Few Shot Approaches

Similarly, we run the zero, one, and few show approaches, keeping the choice of engine constant These graphs as well as a summary table showing scores or engine choice and input example count are shown in 4.1.1. We use the "article-summary" strategy since it is structured most logically to accommodate for few-shot learning as such.

```
article: ...
summary: ...
###
article: ...
summary: ...
###
...
###
article: ...
summary:
```

3.2.3 Testing Prompt Design

For prompt design, we tested prompts that OpenAI claimed worked best for summarization. They explain that "the easiest way to create a summary is to just add the phrase 'tl;dr:' to the end of a document". We also tested the format where "article: " prefixes each article and "summary: " prefixes each summary.

3.2.4 Testing Variable Prompting

We try and measure GPT's responsiveness to requests for summaries of specific word and sentence counts. For this test, we set the max token count returned as 32 as to accommodate different numbers of sentences and words. Here, we graph and find averages for the sentence and word count of each test.

3.2.5 Human Evaluation

Our human evaluation is based on the approach of Song et al. [6] where annotators are asked to assess the informativeness, grammaticality, and truthfulness on a scale of 1 to 5 (where five is "best") of GPT's summaries. We exported 100 of GPT's predictions for the few (10) shot approach on the Gigaword Dataset and had three human evaluators score the model. These results, as well as the results of Song et al. [6] are shown in a table in 4.1.5

3.3 Approach Daily Mail/CNN

The code structure here resembles that for the Gigaword tests. For count of tokens returned, we set max_tokens as 100.

3.4 Limitations

3.4.1 Token Costs

One of the specific limitations this research had was the significant cost of using GPT-3. Given only 300,000 DaVinci equivalent tokens, we had limited resources to debug, test, and compile relevant results for this project. These prohibitive costs also raise the question of whether GPT-3 can be commercially used in products while still being profitable. For the purposes of this study, we had to focus on the GigaWord given it has the shortests inputs (31.4 tokens) and summaries (8.3 tokens). Our tests on Daily Mail/CNN are limited because the dataset is significantly larger.

3.4.2 Cumulative Token Limit for Input

Another constraint we had was the input/output token count limit for the API. OpenAI's API limits the token count sum between the request input and the output to 2048 tokens, with each token on average being the equivalent of four characters. As the character count for an input for Daily Mail CNN sometimes exceeded 3000 characters, it was difficult to run few-shot tests for the dataset.

4. Experiment

We describe the results of the experiment here.

4.1 Gigaword

First, we describe the results for the Gigaword dataset. Table 1 compares GPT's best performing test which takes a few shot approach and uses the Davinci engine and compares the results with the SOTA, drawn from NLP-progress, a github repository that tracks the SOTA for NLP [1]. The f-measure of recall and precision for n-grams is the metric displayed.

Table 1: Comparison with State of the Art:

Model	ROUGE-1	ROUGE-2	ROUGE-L
GPT-3 Davinci 10 shot	26.63	8.08	24.22
ControlCopying (Song et al., 2020)	39.08	20.47	36.69
ProphetNet (Yan, Qi, Gong, Liu et al., 2020)	39.51	20.42	36.69
UniLM (Dong et al., 2019)	38.90	20.05	36.00

Table 2 displays the ROUGE-1 f-measure for the cross sections of each engine and each approach (0, 1, and few-shot).

Table 2: ROUGE-1 f-measure Matrix

	Ada	Babbage	Curie	Davinci
0-shot	06.02	05.78	08.08	07.92
1-shot	07.14	08.59	14.28	20.07
10-shot	13.23	14.47	18.34	26.63

4.1.1. GPT-3 Engine and N-Shot Results

In addition to these summary measures, we provide histograms that chart the distribution of ROUGE-1 scores for each of our benchmarks.

The results of the summarization for few-shot learning clearly show the difference in performances between the four engines. Where the three dimensions of success were described with accuracy/performance, cost, and speed, we can really only analyse performance in our figures and then compare their relative price points.

An example of a high performing summarization on few-shot Davinci (Rouge1 precision score of 0.714), an example description was:

> faced with a mushrooming influence-peddling scandal that has rocked the government, president fernando henrique cardoso has opened a high - level probe into the affair in an attempt to stem political fallout.

with the generated summary being:

> cardoso opens probe into influence-peddling scandal

However, Davinci struggles as well, with an example having ROUGE-1 score of 0.1 being:

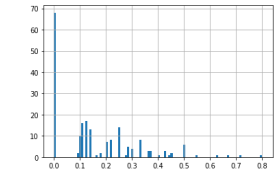
> croatian president franjo tudjman said friday croatian and serb negotiators would meet saturday to thrash out an agreement on the last serb-held area in croatia, under a deal reached at us-brokered talks.

with the generated summary being:

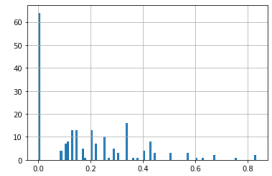
> croatian president hopes deal soon on last serb-

The results of each engine's performance on few-shot learning is displayed in the following histograms.

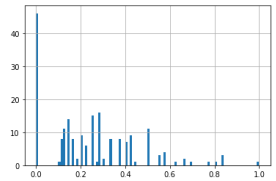
Ada, few-shot



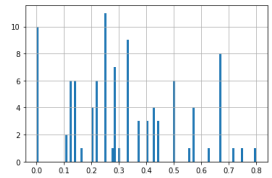
Babbage, few-shot



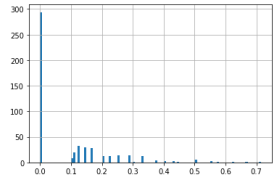
Curie, few-shot



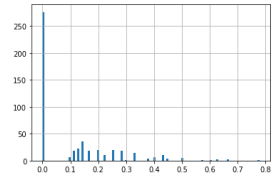
Davinci, few-shot



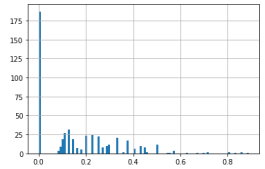
Ada, one-shot



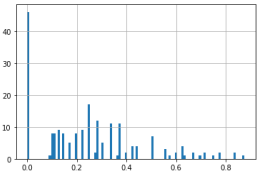
Babbage, one-shot



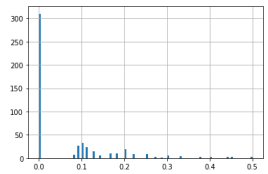
Curie, one-shot



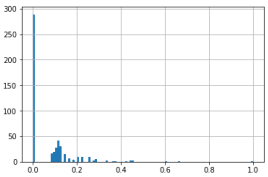
Davinci, one-shot



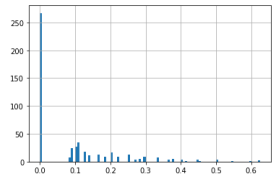
Ada, zero-shot



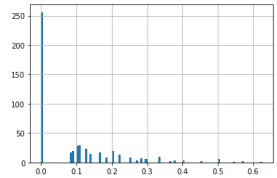
Babbage, zero-shot



Curie, zero-shot



Davinci, zero-shot



The relative performance of the four engines matches the expected output as described qualitatively by OpenAI and also by the relative cost of each token. Davinci, the most expensive and powerful engine, performed well relative to the state of the art ROUGE1 scores. Davinci's performance is no surprise considering its documented success in summarization. The next most powerful engine, Curie, performed less well but still the second most effective. However, not captured in these graphs is Curie's

exceptional speed – which would make its ability to perform these tasks key for its usage as a ChatBot and potentially understanding and summarizing questions in order to find answers. Given Babbage and Ada's focus on simple and fast tasks, as well as cost efficiency, their poorer performance is less surprising especially given that only ten instances were used to train the system on it.

4.2 Strategy performance

Two strategies were studied for summarization: "article-summary" and "tldr". For measuring the performance of these tasks, we had to use the Ada framework due to our tokens limitation and the result was less than impressive.

We found no significant difference in performance. When we actually look at the summaries, however, we can see that prompt design makes a big difference. For the 10-shot, Davinci approach with the "article-summary" strategy, we find, for example, that the summary for the instance:

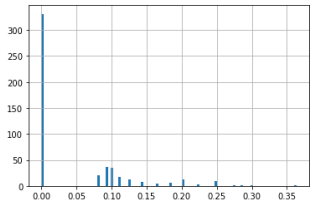
> turnout was heavy for parliamentary elections monday in trinidad and tobago after a month of intensive campaigning throughout the country , one of the most prosperous in the caribbean .

comes out as

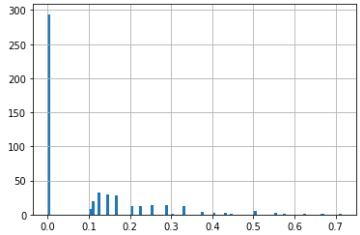
> trinidad holds political elections
> ###
> article: bag

which incorrectly interprets our dividers, which OpenAI recommends you use, as part of the prediction.

Ada engine used with zero shot learning and "tl;dr" summarization prompt



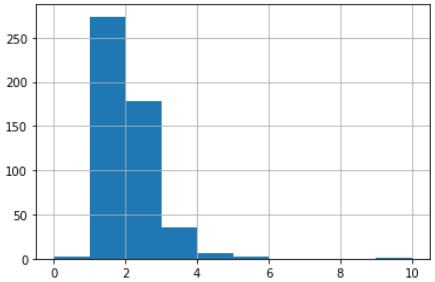
Ada engine used with one shot learning and "article-summary" summarization prompt



4.1. Controlling output

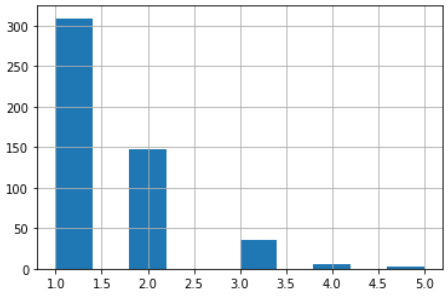
For these tests, we charted the number of sentences produced by OpenAI, using the sent_tokenize function provided by the Natural Language Toolkit or NLTK.

"One Sentence" Strategy # of sentences



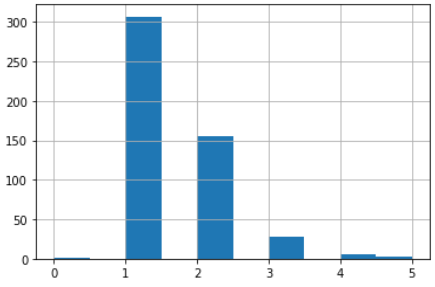
Here, we graph the number of sentences that a "One sentence summary" custom strategy yielded. Whilst a majority of the outputs were still one sentence, there were a significant amount of two sentence summaries as well. We notice that there is not a significant difference when we ask for a "Two sentence summary":

"Two Sentence" Strategy # of sentences



This remained the case even when using the number "2" instead of the word "Two" for our request

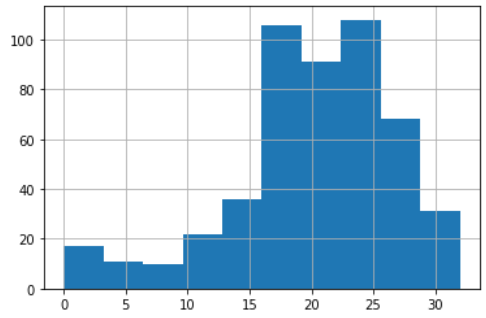
"2 Sentence" Strategy # of sentences



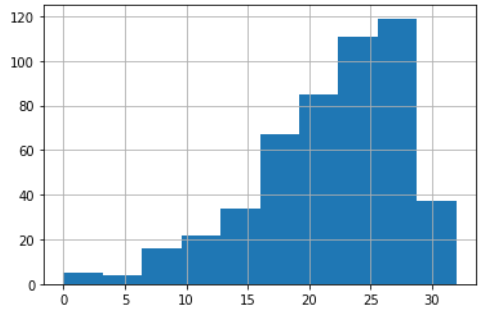
The engine also tested GPT-3 ability to constrain itself to a five word summary vs ten word summary and the distributions show just a slight negative skew for ten word

summary. For the "five word summary:" prompt the summarization still yielded an average of a 20-word response.

"Five word summary:" # of words



"Ten word summary:" # of words



4.1.3 Human Evaluation

We compare our results from human evaluation with the results of human evaluation for other models as provided by Song et al. [6].

Table 4 shows GPT-3 Davinci 10-shot's human evaluated scores as compared to near SOTA models. Song et al. obtained these results having each of 200 instances receive an assessment by five human evaluators. This compares to our human evaluation which had each of 100 instances receive an assessment by three human evaluators.

Table 4: Human Evaluation

	Inform.	Gramm.	Truthful.
GPT-3 Davinci 10 shot	2.404	2.753	2.580
PG Networks	2.768	2.697	2.678
R3Sum	2.748	2.680	2.709
BiSet	2.740	2.634	2.738

4.2 Daily Mail/CNN Evaluation

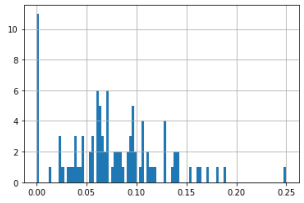
As previously mentioned, due to API token and input/output token length constraints, we could only run tests for the one-shot approach for Daily Mail CNN. These results are shown in Table 5

Table 5: Daily Mail CNN by Engine

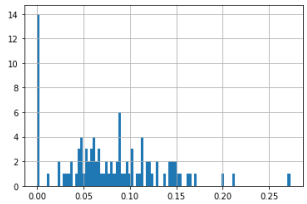
	Ada	Babbage	Curie	Davinci
1-shot	09.78	09.78	08.38	09.05

We display the distributions for the four models respectively.

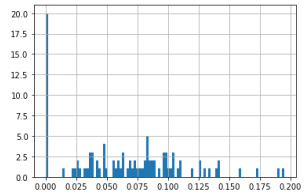
Ada



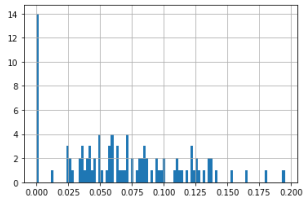
Babbage



Curie



Davinci



5. Conclusion

In closing, we used the ROUGE metric and human scoring to evaluate GPT-3's ability to summarize text and respond to specific summarization commands and strategies. The model compared the performance of OpenAI's four engines: DaVinci, Curie, Babbage, and Ada. These four engines proved to perform relative to each other in terms of their advertised performance, with DaVinci's few-shot summarization task with best performance. None of the other three engines even performed close to the state of the art. Davinci's results with few-shot learning on the Gigaword Dataset was among all of our tests, the one with best performance, yet the model performed just moderately relative to the state of the art.

Running GPT-3 with the "tldr" and "article-summary" strategies, we found no meaningful difference in scoring but human evaluation shows that GPT-3's occasionally fails to understand the prompting format and pattern match accordingly. Testing the "one sentence summary" and "five word summary" strategies, we found that while the

outputs were directionally correct, GPT-3 still struggled with properly executing the prompts, with no real constraint effect realized.

On the Daily Mail/CNN dataset, the model could not reach close to state of the art using the one-shot approach. Unfortunately, here we were constrained by the token limit set by OpenAI for the API, as well the limit for input and output cumulative size, making it impossible to run few-shot tests.

These initial tests were a reminder that natural language processing is still far from reaching "human-level" – and even with a training corpus of 175 billion paramters, GPT-3 still has limitations.

6. References

[1] NLP Progress, Repository to track the progress in Natural Language Processing (NLP), including the datasets and the current state-of-the-art for the most common NLP tasks. https://nlpprogress.com/english/summarization.html.

[2] S. Gupta, K Gupta, Abstractive summarization: An overview of the state of the art, Expert Systems with Applications, 2019. https://doi.org/10.1016/j.eswa.2018.12.011

[3] C. Lin, ROUGE: A Package for Automatic Evaluation of Summaries, Association for Computational Linguistics, 2004. https://www.aclweb.org/anthology/W04-1013/

[4] D. Xiao, ERNIE-GEN: An Enhanced Multi-Flow Pre-training and Fine-tuning Framework for Natural Language Generation, 2020. https://paperswithcode.com/paper/ernie-gen-an-enhanced-multi-flow-pre-training

[5] A. Aghajanyan, Better Fine-Tuning by Reducing Representational Collapse, ICLR 2021. https://paperswithcode.com/paper/better-fine-tuning-by-reducing

[6] K. Song, B. Wang, et. al, Controlling the Amount of Verbatim Copying in Abstractive Summarization, 2019. https://arxiv.org/pdf/1911.10390.pdf

[7] OpenAI, Engines, 2020. https://beta.openai.com/docs/engines