Comparing Pre-trained and Fine-tuned Transformer Models on Patent Data

Cameron Milne
ccmilne@umich.edu

Abstract

Transfer learning offers significant improvements over classical summarization approaches, but for the larger NLP community, little is known about how well smaller variant models can perform on new data. This project compares several of these popular variants before and after fine-tuning on BigPatent, a dataset approximating new scientific language. The results find that fine-tuning significantly improves performance, and that smaller models can be advantageous for NLP practitioners who favor speed over quality.

1 Introduction

Abstractive summarization is one of the most challenging NLP tasks, requiring comprehension of long passages, information compression, and language generation. Transfer learning—where a model is first pre-trained on a data-rich task before being fine-tuned for a downstream task—has offered significant improvements to these challenges in recent years (Yang et al., 2019; Dong et al., 2019; Liu et al., 2019). With self-supervised pre-training on massive datasets, researchers can now fine-tune with relatively little data for a specific task and achieve state of the art results (Devlin et al., 2018; Lewis et al., 2019; Keskar et al., 2019; Raffel et al., 2019). For researchers and NLP practitioners, these methods could offer significant improvements over classical language modeling.

Transfer learning has been embraced by tech companies as the path forward for better performing NLP tools. Companies such as Google, Facebook, and OpenAI have invested heavily into training and releasing large-scale models for public use. Facebook's BART (Lewis et al., 2019), Google's Pegasus (Goodwin et al., 2020), and Google's T5 (Raffel et al., 2019) are among several hundred models capable of producing complex, human-like summaries for written texts. HuggingFace, a community of NLP researchers, has made these models available for use through their API, and researchers, organizations, and other NLP practitioners can now integrate these models into their own applications.

However, model selection remains an open topic. Transformer models are novel technologies, and NLP practitioners looking to benefit from these advances will require clearer guidance on model selection and fine-tuning capabilities than those offered currently in academic journals and documentation. Moreover, the distillations of popular models—versions of BART, Pegasus, and T5 that have been shrunk for faster inference speed—offer even less guidance. This project aims to offer an introductory look at Transformers by comparing the performance of several popular lightweight models before after fine-tuning on a dataset with unseen vocabulary.

2 Project Goals

This project is centered on the following research questions:

- Which pre-trained models can best generalize outside of their domain-specific text summarization task for scientific language?
- How does fine-tuning affect the performance of pre-trained models on a domain-specific summarization task?
- What are the benefits of using small models over large models?

Exploring the summarization capabilities of BART, Pegasus, and T5 on BigPatent can provide a sense of how well these models can generalize outside of their vocabulary training. As more NLP researchers learn the benefits of these models and their applications, understanding which model to

choose and what results can be obtained from fine-tuning can save time and resources when conducting trial and error. The runtime measurements can similarly offer NLP practitioners a sense of how distilled models can help with inference speed downstream.

The project may also yield additional insights into NLP automation tasks within the patent field and identify unique properties of the patent dataset that enable or challenge summarization capabilities. Lastly, this project will assess the viability of several quantitative evaluation metrics for text summarization, revealing which are most useful for this setting.

3 NLP Task Definition

The NLP task is summarization. Both the fine-tuned and pre-trained models will accept a patent article (labeled "description" by the Dataset class) as an input and produce a summary as an output. For modeling purposes, the output will be benchmarked against the true abstract. The performance of these transformers will be evaluated through quantitative methods such as ROUGE, BLEU, and Perplexity. The result of this project will indicate how major Transformer models perform on scientific language, the tradeoffs between smaller and larger models, and the strengths of fine-tuning.

4 Data

BigPatent is the largest scientific-language dataset available for abstractive summarization. Released in 2019, the dataset consists of 1.3 million U.S. patent documents collected from Google Patents Public Datasets using BigQuery (Sharma et al., 2019). BigPatent is significantly larger than other datasets such as the CNN/DailyMail (See et al., 2017), Newsroom (Grusky et al., 2018), XSum (Narayan et al., 2018), arXiv (Clement et al., 2019), and PubMed (Gu et al., 2022) datasets. This project uses a subset of the patents relating to electricity (subset H) which will offer new vocabulary for the pre-trained models. The median description length for Subset H is approximately 2500 words and the median abstract length is 109 words. The BigPatent dataset is available through the Datasets class and HuggingFace's Transformer libraries are designed to accommodate the Datasets datatype.

5 Related Work

Classical approaches to summarization have been extractive approaches, utilizing *n*-gram models such as TF-IDF (Christian et al., 2016) and Bayesian models (Nomoto, 2005). Machine learning upgraded these approaches with Seq2Se2 models and encoder-decoder frameworks (Nallapati et al., 2016), but the introduction of the Attention layer in encoder-decoder models, which weighs words differently in order to better understand latent semantic meaning, allowed for the development of today's pre-trained models (Vaswani et al., 2017). The release of Transformer models such as GPT, ULMFiT, and ELMo in 2018 along with XLNet, RoBERTa, ALBERT, Reformer, and MT-DNN in 2019 have quickly advanced many NLP tasks, but the rate of progress has exceeded the pace of research evaluating these models and their applications. Thus, research has been limited regarding the scope of this project.

Some studies have explored the effects of fine-tuning, such as the vocabulary lost during fine-tuning (Chen et al., 2020), the steps necessary for learning a task (roughly five orders of magnitude lower than the parameter count for a model) (Radiya-Dixit and Wang, 2020), and the number and location of layers that absorb most of the fine-tuning (Wortsman et al., 2021; Radiya-Dixit and Wang, 2020). Researchers have also noted that distinct random seeds can lead to substantially different results, prompting the need for measures to stop training when results are less promising earlier (Dodge et al., 2020), and most importantly for this project, that performance on representations of out-of-domain sentences are weak (Radiya-Dixit and Wang, 2020).

Research on the use of small models is similarly limited. Researchers have devised strategies for compressing models while preserving state of the art results. One strategy, "Shrink and Fine-Tune" (SFT), was used to distill large models like BART and Pegasus into smaller versions with faster inference speeds (Shleifer and Rush, 2020), critical breakthroughs for expanding access and usability of these models for everyday NLP researchers.

Researchers have struggled to develop innovative evaluation metrics for summarization tasks. While the Pyramid method (i.e. manual annotation) is the gold-standard approach for evaluating text summarization tasks (Nenkova and Passonneau, 2004), annotation is costly and often requires

more time than available. Researchers have therefore relied on scalable evaluation metrics. In 2002, BLEU (Bilingual Evaluation Understudy Score) was proposed as a precision-based metric for evaluating the quality of generated texts (Papineni et al., 2002). BLEU counts the number of words that appear in the generated text as well as the reference text, divided by the length of the reference. The researchers realized this could be cheated if a generated text repeats a word enough, so they implemented a brevity penalty that penalizes terms that occur more frequently in the machine-produced text. In 2004, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) was introduced and has since become the most popular evaluation method for summarization (Lin, 2004). ROUGE operates similarly to BLEU, but also offers analysis of precision, recall, and F-scores which are reported more often. The effectiveness of ROUGE has been reviewed as technologies have evolved, and researchers have found that the conclusions from older datasets do not necessarily hold true on modern datasets and systems (Bhandari et al., 2020). However, without many summarization evaluation methods available, BLEU and ROUGE can offer a quantitative perspective on the quality of generated summaries. Lastly, researchers have proposed a metric called Perplexity which attempts to evaluate the coherence of a summary. Because a language model is itself a probability distribution over entire sentences or texts, Perplexity can assess the accuracy of word predictions within text generation tasks, offering a large-scale substitution for human annotation.

6 Methodology

Model selection required considerable thought to ensure as much standardization as possible, while providing enough differentiation to address critical components of Transformer architectures. The baselines consist of an industry standard Lead-3 and Pegasus's BigBird model which has been fine-tuned for BigPatent by researchers at Google. The models being studied are BART, Pegasus, and T5, three of the most prominent conditional language generation models. For each of these models, a small variant was selected. For BART and Pegasus, distilled versions are available for use. These models were trained on XSum, a dataset built by harvesting 230,000 online articles from the BBC for summarization tasks (Narayan et al., 2018). T5 was trained on the C4 dataset, a collection of approximately 750GB of English-language text sourced from the public Common Crawl web scrape. All datasets used in the training of pre-trained and fine-tuned models are in English (Raffel et al., 2019). The models are explained below in detail.

6.1 Baselines

The first baseline summarizer is Lead-3 which takes the first three sentences of a patent's description. The second baseline will use BigBird, a sparse-attention based transformer which extends Transformers-based models such as BERT for longer sequences which has been fine-tuned on BigPatent (Zaheer et al., 2020). The generated summaries from BigBird will be longer and more sophisticated than the smaller models, likely even after training. BigBird accepts an input sequence of 4098 tokens, which should allow most of the BigPatent articles to be read completely. One of the core limitations of Transformer-based models is the quadratic dependency (memory) on the sequence length which BigBird remedies by reducing the dependency to linear. BigBird will provide another perspective by demonstrating what smaller models can aspire to reach.

6.2 BART

BART (Bidirectional and Auto-Regressive Transformers) is pre-trained on document rotation, sentence permutation, text-infilling, and token masking and deletion objectives (Lewis et al., 2019). The variant of BART used for this project was *distilbart-xsum-12-1*, a distilled version of BART with just 222 million parameters and the fastest inference time of any distilled BART at 90 milliseconds (Shleifer and Rush, 2020). BART accepts an input sequence of 1024 tokens.

6.3 Pegasus

PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive SUmmarization Sequence-to-sequence) was designed for abstractive summarization and pre-trained with a self-supervised gap-sentence-generation objective (Goodwin et al., 2020). The variant of Pegasus used for this project was *distill-pegasus-xsum-16-4*, a distilled version of Pegasus with 369 million parameters and an inference speed of 2038 milliseconds (Shleifer and Rush, 2020). Pegasus accepts an input sequence of 512 tokens.

6.4 T5

T5 (Text-to-Text Transfer Transformer) is pre-trained on several unsupervised and supervised objectives, such as token and span masking, as well as translation, classification, reading comprehension, and summarization. The variant of T5 used for this project was *t5-small*, a version of T5 with just 60 million parameters (Raffel et al., 2019). T5 accepts an input sequence of 512 tokens.

6.5 Fine-tuning the Summarizers

Fine-Tuning BART, Pegasus, and T5 was possible via HuggingFace's Trainer class which supports distributed training on multiple GPUs. Because each model varied in size and memory capabilities, batching was adjusted to optimize for speed. For each model, a length penalty of 2.5 was used to encourage longer summaries, and the number of beams was set to 8.

6.6 Evaluation Metrics

Three evaluation metrics are applied: ROUGE, BLEU, and Perplexity. ROUGE works by comparing a generated summary against a reference summary, computing the precision and recall of the generated summary by looking for overlapping words.

$$Recall = \frac{\text{number of overlapping words}}{totalwordsinreferencesummary}$$

$$Precision = \frac{\text{number of overlapping words}}{\text{total words in generated summary}}$$

The value of recall and precision in a summarization task is that it captures the nuances of various possible summaries that can be effective. Machines might produce summaries that vary in word choice or in word order, so precision and recall allow different possibilities. Additionally, ROUGE provides a f-measure (i.e. F-Score) option for analyzing the harmonic mean of the precision and recall scores. The equation is below:

$$F - Score = \frac{\text{precision * recall}}{\text{precision + recall}}$$

For the results, the average ROUGE-1 and ROUGE-L scores across all summaries are reported in the results. ROUGE-1 measures unigrams whereas ROUGE-L measures the longest matching sequence of words.

BLEU (bilingual evaluation understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality here is defined as the similarity of a machine's output to a human's. BLEU works by counting matching n-grams in the generated summary to the n-grams in the reference summary, where a unigram would be each token and word order is ignored. Matching n-grams counts are penalized to ensure that the occurrence of words in the reference summary are noted such that the generated summary doesn't get away with using more than one match.

The equation below defines the process at work with BLEU. First, the geometric average of the modified n-gram precision scores, p_n, using n-grams up to length N and positive weights w_n summing to one. Next, let c be the length of the generated summary and r be the reference summary corpus length. We compute the brevity penalty BP:

$$BP = \begin{cases} 1 & if\ c > r \\ \frac{1}{e^{(1-r/c)}} & if c <= r \end{cases}$$

Then,

$$BLEU = BP * exp\left(\sum_{n=1}^{N} w_n log p_n\right)$$

Perplexity is also applied to provide an approximation of a model's ability to produce readable sentences. From the test set, a unigram model was established by adding each word w and its probability p of appearing in the dataset (words with low probabilities received a value of 0.01 for smoothing). Then, for each generated summary, the perplexity value is calculated by calculating the inverse probability of the words appearing in the generated summary:

$$PP(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, ..., w_N)}}$$

Generated summaries that are considered more readable and syntactically correct will be given lower scores.

	ROUGE		BLEU		
Models	**R-1**	**R-2**	**BLEU**	**Length Ratio**	**Perplexity**
Lead-3	28.6	18	4.616	**73%**	**779**
BigBird	**31.8**	**22.2**	**5.142**	6%	863
BART	20.2	14.6	0.267	25%	**391**
Pegasus	**24.1**	**16.7**	**1.284**	**35%**	528
T5	10	8.3	0.003	11%	680
$BART_{trained}$	**35.1**	**24**	**4.405**	**46%**	**562**
$Pegasus_{trained}$	28.5	21.6	3.297	45%	740
$T5_{trained}$	13.2	11.2	0.006	11%	816

Table 1: Results: Evaluation Metrics

	Pre-Trained		Fine-Tuned	
Models	**Minutes**	**Samples/Sec**	**Minutes**	**Samples/Sec**
Lead-3	2	119.0		
BigBird	174	1.4		
BART	18	13.2	23	10.3
Pegasus	23	10.3	25	9.5
T5	**7**	**34.0**	**9**	**26.4**

Table 2: Runtime Results

7 Results

The results for the baselines, pre-trained models, and fine-tuned models are displayed in *Table 1*. The best results for each metric are highlighted within each category of models (i.e. baselines, pre-trained, fine-tuned).

Overall, BigBird outperforms the smaller models (before fine-tuning) in ROUGE-1, ROUGE-L, and BLEU as expected. Lead-3 has the longest average generated summary, likely influencing higher ROUGE and BLEU scores. Within the pre-trained models, Pegasus outperforms BART and T5 in every metric but perplexity. However, $BART_{trained}$ outperforms both $Pegasus_{trained}$ and $T5_{trained}$ within the fine-tuned models, suggesting BART might be better capable of learning scientific language than Pegasus. $BART_{trained}$ even outperforms BigBird in ROUGE and Perplexity, a surprising finding given BigBird could serve as the industry standard for the BigPatent dataset and accepts a longer input sequence. T5 underperforms both before and after fine-tuning in every category, indicating the model is a weaker choice for this task.

Runtime results are displayed in *Table 2*. T5 can generate the most summaries per second at 34. After fine-tuning, all three summarizers slow down slightly, with a difference of about 5 minutes for BART and 2 minutes for both Pegasus and T5. BigBird is the slowest summarizer at approximately 1.4 samples per second.

8 Discussion

Several observations from Table 1 and Table 2 are worth noting:

(1) The length of generated summaries significantly impacted ROUGE and BLEU results. Higher ROUGE and BLEU scorers, such as Lead-3 and BigBird, produced higher average length ratios than lower scorers like T5. Because ROUGE and BLEU are metrics for evaluating overlapping n-grams, it's reasonable that longer summaries will likely have more overlapping words with reference summaries. **(2) BART and Pegasus saw an increase in average length ratio after fine-tuning, while T5 remained the same (even with an increase in ROUGE scores)**. This change occurred without enforcing any new length penalties, a pattern worth exploring in future research. The takeaway here is that fine-tuning somehow enabled these models to begin recognizing new terms that could be used in generating a summary. **(3) T5's poor performance before and after fine-tuning relative to BART and Pegasus could be explained by different underlying pre-trained datasets**. T5 was trained on random text scraped on the internet while BART and Pegasus were pre-

trained on XSum, a collection of news articles. XSum might have provided BART and Pegasus with clearer patterns for assessing semantic meaning. **(4) Larger models are considerably slower**. Runtime differences between BigBird and the three smaller models were notably large; BigBird could only produce 1.4 summaries per second whereas the BART, Pegasus, and T5 were able to produce 10.3, 9.5, and 26.4 summaries per second. This can be explained by several model characteristics such as varying accepted input sequence lengths, number of transformation layers, and more, but the takeaway for NLP researchers is that smaller models can approach the level of quality capable of bigger models and preserve their fast inference speeds. **(5) Perplexity appears to be correlated with the average length ratio**; fine-tuning led to an increase of approximately 200 across all three models. BART scored the lowest in perplexity before and after fine-tuning, suggesting BART's summaries are more understandable.

9 Conclusion

Results indicate fine-tuning yields significant improvement across ROUGE and BLEU metrics. For NLP practitioners or researchers interested in obtaining high quality summaries, these results demonstrate that a smaller model with less parameters can perform just as well as larger models. Where inference speed is prioritized, fine-tuning on small models promises good results and preserves much of the speed advantages.

One consideration of using smaller models, however, is that generated summaries are shorter on average. BigBird's average length ratio was 60% whereas $\text{BART}_{\text{trained}}$ and $\text{Pegasus}_{\text{trained}}$ were 46% and 45% respectively, even with substantial increases in ROUGE scores. Furthermore, input sequence length is a key characteristic of these models. BART's improvement over Pegasus after fine-tuning could be influenced by BART's ability to accept twice as many words (4096 over 2048). Inference speeds appear to be less influenced by this quality as well.

10 Other Things Tried

Two additional baselines were experimented with before deciding they weren't needed. One was a random summarizer, which took random words from an article until it reached the same number of words as the corresponding abstract. The problem with including this was that it achieved higher ROUGE scores than the smaller models before and after fine-tuning, because it was the only model that could achieve a perfect length ratio. The second baseline dropped from the project was a TF-IDF summarizer, which produced sentences that were only slightly more comprehensible than the random summarizer. As Lead-3 is the industry standard for summarization baselines, this was the only baseline kept.

Modifications during fine-tuning were also experimented with in order to generate the longest possible summaries. Because these smaller models accept fewer words in the input sequence, they generate smaller summarizations. Controlling for parameters such as min length, length penalty, and more didn't yield longer summaries that were also comprehensible. In fact, the model would "cheat" by repeating words in the case of BART, and repeating subsequences in the case of Pegasus.

11 Future Opportunities

Limitations of this study include a limited comparison of models. Distilled versions of three popular models can offer a sense of which models are best equipped for the BigPatent dataset, but there are hundreds of models available built on different datasets. Future research might look into a greater diversity of models trained on different datasets (BART and Pegasus were both trained on XSum, a corpus of BBC articles).

Future research could also investigate alternative evaluation metrics. Overlapping *n*-grams are still a weaker assessment for language generation tasks. This project could have benefitted from a self-annotation sample where generated summaries could be ranked on a Likert Scale for fluency, accuracy, and more.

References

Manik Bhandari, Pranav Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting.

Hans Christian, Mikhael Agus, and Derwin Suhartono. 2016. Single document automatic text summarization using term frequency-inverse document fre-

quency (tf-idf). *ComTech: Computer, Mathematics and Engineering Applications*, 7:285.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation.

Travis Goodwin, Max Savery, and Dina Demner-Fushman. 2020. Flight of the PEGASUS? comparing transformers on few-shot and zero-shot multi-document abstractive summarization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5640–5646, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding.

Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016. Sequence-to-sequence rnns for text summarization. *ArXiv*, abs/1602.06023.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Tadashi Nomoto. 2005. Bayesian learning in text summarization.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Evani Radiya-Dixit and Xin Wang. 2020. How fine can fine-tuning be? learning efficient language models.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization.

Sam Shleifer and Alexander M. Rush. 2020. Pre-trained summarization distillation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2021. Robust fine-tuning of zero-shot models.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences.