SI 699 Big Data Analytics

Final Project Report

Evaluating the Performance of Text Summarization Models Across Domains

Michael Kalmus
Cameron Milne
Harrison McCabe

April 21, 2022

ABSTRACT

Our project investigates the generalizability of four Transformer models fine-tuned for text summarization. We assess the usefulness of these models by applying them to different datasets and domains they were not originally fine-tuned on. We believe this methodology could prove useful in cases where pre-trained models do not exist or customizing a model would be infeasible. This application is unique when compared to most existing literature which focuses on fine-tuning and evaluating models within their training domains. To accomplish our research goal, we generate summaries with each model and a simple extractive baseline on news articles, scientific papers, and patents. We then compare metrics that relate a generated summary to a gold standard summary. The results show that complex models applied on datasets they were not fine-tuned on produced worse summaries than those produced by a simple extractive baseline.

EXECUTIVE SUMMARY

- **Context and Importance**: In this paper, we study the performance of state of the art text summarization models with an emphasis on evaluating models outside of the domains they were trained and fine-tuned on. Effective summarization is becoming increasingly useful for researchers, organizations, and other Natural Language Processing (NLP) practitioners and can help save time and resources to supplement instances where long blocks of text must be stored or processed.

- **Previous Work and Open Questions**: Previous research has focused on the mechanics of Transformer architectures, the effects on fine-tuning, and the slow progress of summarization evaluation metrics. Furthermore, most work focuses on fine-tuning and evaluating models on the same datasets which does not assess how they perform in different domains, bringing to light the following questions:
    - How generalizable are pre-trained Transformers for summarization?
    - Which summarizers work best in particular spaces?
    - How do the performances of transformer-based summarizers compare to that of a non-neural-based summarizer?

- **Data**: We studied data from three domains: Patents (Big Patent), scientific papers (PubMed), and the news (CNN/DailyMail). Datasets were chosen because they are commonly used and fundamentally different in terms of article length and level of abstraction in summaries.

- **Methods**: We applied four Transformer models, BigBird; PEGASUS; BART; and T5 to the three datasets. Generated summaries were then compared to ground truth summaries in terms of their overlap (ROUGE and BLEU), perplexity, readability, and runtime. The models were chosen because they have past success in summarization and all facets of implementation from dataset loading to evaluation were completed with the Hugging Face ecosystem and specifically the Datasets, Tokenizers, Transformers, Metrics libraries.

- **Findings**: After reviewing relevant literature and evaluating our models across domains, we arrived at the following key conclusions:
    - Pre-trained models do not generalize effectively outside of their training domain.
    - Dataset comparisons are challenging: structural differences between data from different domains present difficulties for models with fixed input sequence lengths.
    - Fine-tuning is likely necessary for models to be used in production.

- **Limitations**: Findings may have been limited or challenged by the following points:
    - Inherent differences between models: while standardization was applied where possible, our model families come with varying numbers of transformation layers, input sequence lengths, and underlying datasets for which the model was trained on, making comparison difficult when models are fed varying context.
    - Dataset differences: While CNN/DailyMail might offer notable points in the first paragraph, PubMed's notable points might be buried in the conclusion, challenging the models that accept smaller input sequence lengths.
    - Limited scope of comparison: only four models were examined for comparison across three datasets while hundreds exist. However, we chose four distinct models with summarization use-cases, and provide a framework for integration with other models.

1. INTRODUCTION

As the amount of raw, unstructured text data available to researchers continues to grow, so does the need to summarize text both efficiently and effectively. Furthermore, the availability of increased computing power and open-source libraries provides researchers with the tools needed to study summarization at deep levels. In this paper, we present the methodology and results employed in applying modern summarization techniques across text from different domains. We also provide readers with links to our code[1], with the hopes that this work can be used and studied further. In completing this project, our team identified the following objectives:

1. Study and evaluate current tools and techniques used in text summarization
2. Apply state of the art Transformer models in different domains (news, patents, scientific papers) without without fine-tuning them for the specific dataset
3. Evaluate differences in performance between models and domains

The Transformers[2] library, a unified API that provides access to hundreds of pre-trained models, has helped researchers use and repurpose these advanced models for their own applications (Wolf et. al, 2019). These models can be fine-tuned for many different Natural Language Processing (NLP) tasks including classification, named entity recognition, and question-answering. Transformers also offers a range of methods for summarization such as frequency-based and model-based approaches (Wolf et. al, 2019); these methods will be the focus of this project.

A summary can either be extractive or abstrative: in extractive summarization, sentences are used word-for-word, while in abstractive summarization, words not in the original text may be used. Generally, abstractive summarization is a more difficult task because abstracts resemble human-created summaries and capture context and semantics of language (Zhang et al., 2019). While many techniques and tools exist, we did not find a unified approach comparing state of the art models across domains and datasets, and this is the topic we aim to study. This is notable because different researchers use different methods to fine-tune models and preprocess data which often does not provide for direct comparison between model implementations. Furthermore, code may not accompany current research, and we aim to provide clear documentation and code on how we implemented the models in Python. Lastly, little research exists studying how models fine-tuned on one dataset generalize to other datasets and domains.

Practically, text summarization saves users time and costs where long documents of text must be read or processed. Where documents need to be stored, summarization can also save storage space on disks by saving summaries rather than full texts. Therefore, building an efficient text summarization tool can offer time savings to users who want to comprehend text faster and cost savings to corporations who need to store or process large amounts of text. By comparing state of the art models across domains, we could facilitate the implementation of domain-specific applications like a news summarizer. We also hope that by providing our code and documentation, our research and approach can be extended to other models or domains.

Our project uses four Transformer models, BigBird (Zaheer et al. 2020), PEGASUS (Goodwin et al. 2020), BART (Lewis et al. 2019), and T5 (Raffel et al. 2019) for summarization and compares their summaries for patent data, news data, and scientific paper data. Of the four Transformers, BigBird was the only one specifically fine-tuned for one of the datasets used; other models were trained with similar data (i.e. same domain but different data) or on data from a different

[1] https://github.com/mkalmus/transformer_evaluation_for_summarization
[2] https://github.com/huggingface/Transformers

domain. Results are compared with a non-Transformer baseline in which a summary is created as the concatenation of the first three sentences in a piece of text, a baseline approach taken by other researchers (Lewis et al., 2019). These models were chosen because previous research found success using the architectures specifically for text summarization in different domains as discussed in their respective papers. The following two sections explain the rationale behind the chosen models and datasets in further detail. Moreover, we include code which can generalize the full summarization pipeline to any pre-trained model available from the Transformers package with associated tokenizer and model weights. We hypothesized that state of the art Transformers would outperform the simple baseline significantly, and considered accepted performance measures like ROUGE scores (Nallapati et al., 2004), as well as runtime in our analysis. Ultimately, the results showed that our hypothesis was wrong, and that the baselines consistently outperformed every Transformer across all datasets, with the exception of BigBird on the patent data for which it was specifically fine-tuned. The CNN/DailyMail dataset performed slightly better with Transformers, but the generated summaries were much smaller than expected. Our results conclude that the pre-trained Transformers models evaluated, without fine-tuning, are insufficient for summarization tasks in new domains. We also conclude that fine-tuning can significantly improve results at the cost of increased runtime.

2. PROBLEM DEFINITION AND DATA

The first step in our research was to formally define the problem and research direction and to acquire data that could be used to study text summarization. We ultimately chose to study four distinct Transformer models and their performance across three domains. In this section, we expand upon our motivation for this research and rationale in choosing the different models and datasets.

2.1. Problem Definition

As researchers are very quickly creating more ways to use Transformers to summarize text, there has been minimal research comparing how different pre-trained models perform different types of text without fine-tuning, especially with respect to the widely-used models from the Transformers library. In practice, most summarization models in the Transformers library are trained on news data, with generated summaries compared against human-generated 'gold standards'[3]. However, the applications of text summarization go beyond news articles and pre-trained models may not be available for the dataset of interest. To provide one example, the BigPatent dataset on the Hugging Face hub has only one pre-trained model which has been downloaded over 3,500 times in April 2022 alone[4]. In this case, users must choose the one model, fine-tune a different model on the patent data, or use a pre-trained checkpoint from a different domain. The latter approach can provide significant time savings as no time will need to be spent fine-tuning and users can choose models of different sizes. This means that these models will need to be generalizable to domains with different structures of information presentation in order to be practical. Therefore, this research seeks to answer the following questions:

- How generalizable are pre-trained Transformers for summarization?
- Which summarizers work best in particular spaces?
- How do the performances of transformer-based summarizers compare to that of a non-neural-based summarizer?

For our project, we evaluate the performance of four different Transformer architectures and a

[3] https://huggingface.co/models?pipeline_tag=summarization&sort=downloads
[4] https://huggingface.co/google/bigbird-pegasus-large-bigpatent

three-sentence baseline across three domains including patents, scientific papers, and news. The domains were chosen because large-scale labeled data exists within each domain as further discussed in the next subsection.

Specifically, the four Transformer architectures used were BigBird (Zaheer et al. 2020), PEGASUS (Zhang et al. 2020), BART (Lewis et al. 2019), and T5 (Raffel et al., 2020) , and a three-sentence baseline consisted of the first three sentences of each article joined by a newline character. The models were chosen due to their use cases in summarization as discussed in their respective papers and because pre-trained models existed for the summarization use-case. Additionally, each model represents a different "family" of models, which differ in terms of training methodology, tokenization schemes, and architecture. Furthermore, all models except BigBird were studied in out-of-domain applications, and using one fine-tuned model gave us another comparison point that should perform better when evaluated on the dataset it was fine-tuned with. In the Related Work section, we outline some of the differences and results for these models with respect to summarization.

This paper will not explore all aspects of the Transformer architecture in detail and we instead direct interested readers to the respective papers for each model and to the general Transformer architecture first proposed in "Attention Is All You Need" (Vaswani et al. 2017). Therefore, this paper is not a discussion or study on the theoretical and mathematical differences between different Transformer architectures, but rather an empirical study on the effectiveness of different widely-used models for summarization outside of the domain they were originally fine-tuned for.

Success in this project will come from displaying clear and understandable results on how each model performs with each dataset. Additionally, finding efficient ways to perform all aspects of summarization from data loading to evaluation was needed to achieve success as summarization with Transformers is a highly intensive task (Vaswani et al. 2017). Two quantitative evaluation metrics, ROUGE (Lin, 2004) and BLEU (Papineni, 2002), provided insight into the quality of the summaries. Our methods to efficiently process the data and further discussion on the metrics used are discussed later in the Methodology section.

At the end of this research, we offer discussion on how common pre-trained models available through the Transformers library perform outside of the domain they were fine-tuned for and remark on both summary quality and human readability.

2.2. Data

Since our research consisted of creating abstractive summaries from multiple domains, we utilized data from three different domains: patents, scientific papers, and the news. To study abstractive summarization, we first needed texts that had "gold standard" summaries and evaluated multiple datasets created for this use-case. We wanted datasets that were fairly large in size and from fundamentally different domains, and found the BigPatent dataset (Sharma et al. 2019), a dataset of PubMed articles and abstracts (Cohan et al. 2018), and the CNN/Dailymail dataset (Nallapati et al., 2016) to be sufficient due to their diversity and availability as described later in this section. Raw data was accessed through the Datasets package[5] in Python. In the Related Work section, we describe other work done with these datasets.

Each dataset contains a full-text (i.e. a patent, scientific paper, or news article), as well as a human-created abstract for each piece of text. An abstract contrasts an extractive summary because the sentences used may not directly come from the text (Zhang et al., 2019). Table 1 below shows some summary statistics of the test partition of each dataset. Test partitions were used because they were created for the purpose of model evaluation; however, we did verify that the models could be run

[5] https://pypi.org/project/datasets/

in a timely manner across the larger training data as discussed in the Other Things We Tried section. Due to the many steps involved in processing text for modeling and evaluation, as well as the length of the full-texts, operating on any partition can be computationally intensive as long articles must be tokenized, encoded, decoded, and evaluated.

Table 1: Statistics of Each Dataset Used for Evaluation

Dataset	Number of Article/Summary Pairs	Median Article Length (words)	Median Summary Length (words)
BigPatent (Patents)	14,279	2,496	108
PubMed (Scientific Papers)	6,658	2,238	186
CNN/DailyMail (News)	11,490	606	48

As shown in Table 1, there are fundamental differences when comparing document length across datasets. For example, patent texts are more than quadruple the length of news texts when looking at the median article lengths. Practically, this means that a piece of text is more likely to be relevant in the news documents compared to patent documents because less text captures a larger portion of the data. Looking at the median summary lengths, we can see that news summaries are significantly shorter than summaries for the other documents. Conversely, scientific paper abstracts are the longest summaries of all the data. In Figure 01 through Figure 06 on the the bottom on this page and on the following page, we show the distribution of sentences in full-texts (left figures) and summaries (right figures) for all three datasets. All full-texts are plotted on the same scales and all summaries are plotted on the same scales to facilitate comparison between the two subsets.

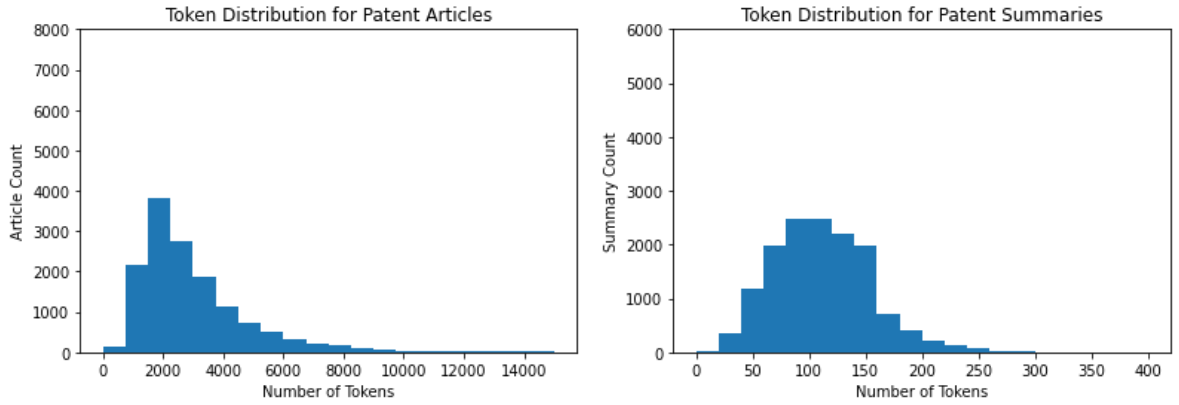
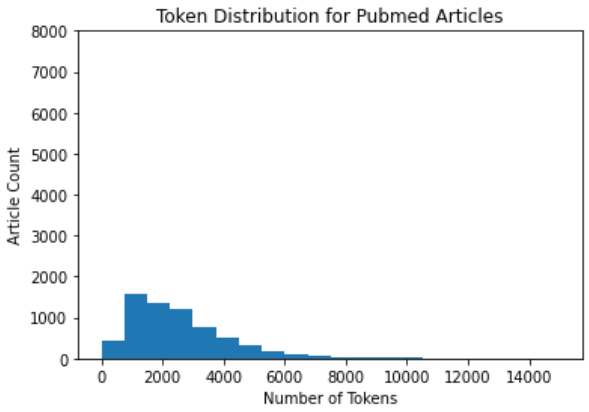


Figure 05 (left) and Figure 06 (right): word/token distributions for patent paper data



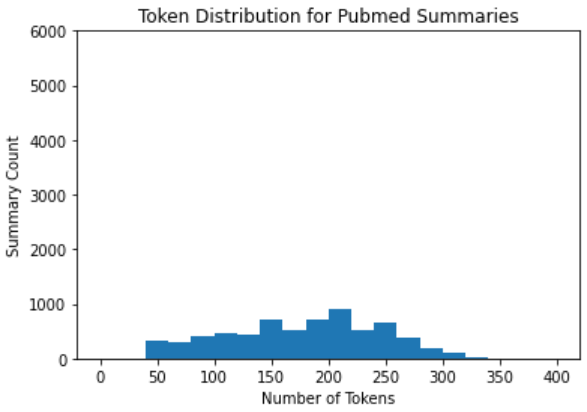
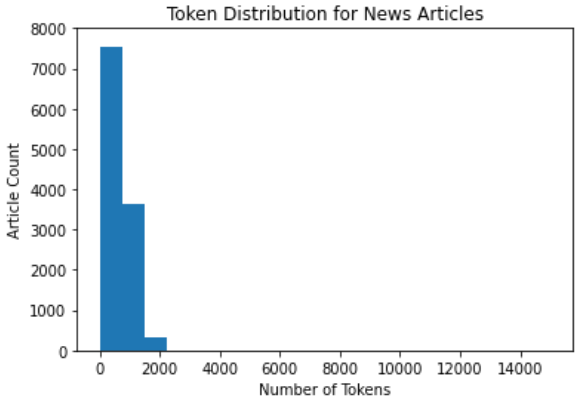


Figure 03 (left) and Figure 04 (right): word/token distributions for scientific paper data



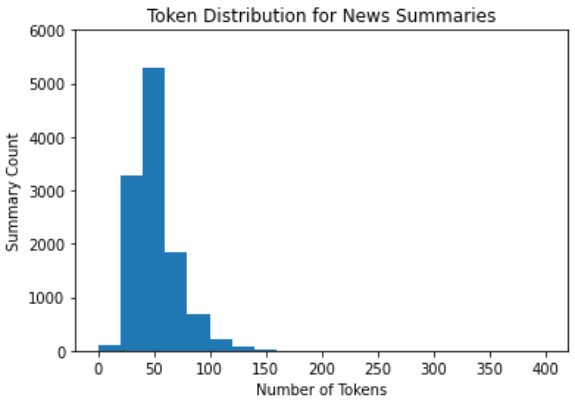


Figure 01 (left) and Figure 02 (right): word/token distributions for news data

Looking at the distribution of article lengths, each dataset's articles appear to be right-skewed, with most articles being shorter and a long tail for longer articles. The range of article length also differs in the same fashion the average article length does, with patent data having some articles up to 15,000 tokens. With regards to the summaries, summaries for news and patents also appear to be right-skewed with shorter summaries, but to a lesser degree than the articles. Scientific paper summaries, however, appear to be less skewed than the other data. We also see that news summaries are far shorter than both patent and scientific paper abstracts, with almost all summaries having less than 150 tokens.

3. RELATED WORK

Summarization is well-studied in the field of NLP and many different models, tools, and techniques are used to complete the task. In studying text summarization, we reviewed relevant literature on different models and their performance with regards to summarization. We also reviewed papers that discuss differences and other work done with the same datasets.

3.1. Comparison of Transformer Models for Summarization

For this research, we leveraged several variants of different Transformer architectures because past work indicated their success and use-cases in summarization (Lewis et al., 2019; Raffel et al., 2019; Zaheer et al., 2020; Zhang et al., 2019). The original Transformer architecture was first proposed in

2017 by researchers at Google Brain (Vaswani et al., 2017) and has since been expanded upon for a wide range of use-cases across NLP including question-answering, summarization, and text classification (Raffel et al., 2019). This paper does not provide extensive discussion on the general Transformer architecture, but rather offers a brief introduction and points interested readers to the relevant papers associated with each model architecture. Moreover, we study and discuss how four different Transformer-based models perform in different domains without fine-tuning.

Each model evaluated represents a different "family" of models where different variants in terms of model parameters and training methodology are publicly available[6]. These Transformer models are pre-trained through a process called Masked Language Modeling (MLM) and vary in terms of the specific MLM used to train them. In MLM, tokens or sentences are dropped, or "masked," from a piece of text, and the model tries to recreate the original text given the remaining sentences. This approach provides models with a method of self-supervision, in which they can be trained without labeled data and fine-tuned for downstream tasks such as summarization (Zhang et al., 2019). Below, we highlight some key points and differences in the pre-training, objectives, and results of the models studied in this research:

- **BigBird** (Zaheer et al., 2020): BigBird represents recent research to extend Transformers to longer input sequences. Most Transformer models, including the other models studied in this research, accept up to 512 tokens of input when preparing summaries due to their quadratic complexity (i.e. O(n^2)) where longer articles result in significantly longer computation time. BigBird introduced a new attention mechanism, allowing the model to accept inputs up to 4096 tokens with linear (i.e. O(n)) complexity. This mechanism allowed the largest variants to outperform PEGASUS, BART, and T5 for summarization when fine-tuned on longer documents but not when fine-tuned on shorter documents.

- **PEGASUS** (Zhang et al., 2019): PEGASUS represented a new approach to MLM and pre-training in which the most important sentences as measured by their similarity with the rest of the text are masked. PEGASUS was originally trained on the same web-crawled documents as T5 (described below), but has also been experimented upon with fine-tuning on different datasets and using different numbers of parameters and different hidden layer sizes. This pre-training method was created with summarization in mind, and the authors show that PEGASUS outperformed other state of the art summarization architectures from that time including T5 and BART when fine-tuned on summarization data in different domains.

- **BART** (Lewis et al., 2019): BART uses MLM for self-supervised pre-training like PEGASUS, but represents earlier work than PEGASUS and BigBird in which the authors mask individual tokens instead of full sentences. Pre-training also consists of creating noise in the training data by permuting sentences, deleting tokens, and similar text manipulation methods. It was originally trained on a combination of Wikipedia data and books for the purpose of summarization, and the authors show that it outperforms a three-sentence, extractive baseline when being fine-tuned for summarization on individual datasets.

- **T5** (Raffel et al., 2019): T5 represents one of the first large-scale studies of a general purpose Transformer architecture that can be used for a variety of NLP tasks including summarization, question answering, and machine translation. T5 treats every NLP task as a "text-to-text" problem, in which we take text as an input and produce text as output, allowing the same

[6] https://huggingface.co/models?pipeline_tag=summarization&sort=downloads

model with the same training objective to be applied for multiple tasks and fine-tuned to work with different datasets. The authors report that without pre-training, T5 performs worse for news summarization than a standard encoder-decoder Transformer architecture but performs slightly better when fine-tuned on summarization datasets. A potential drawback is that training T5 is costly as the smallest variant has 60 million parameters and the largest variant has 11 billion, and other models like PEGASUS have been shown to be as effective while being smaller in size (Zhang et al., 2019).

Comparing across the original papers associated with each model, the BigBird authors show that it achieves better metrics than PEGASUS (Zaheer et al., 2020), and the PEGASUS authors show that their model outperforms T5 and BART (Zhang et al., 2019). However, the results largely show small improvements (i.e. BigBird improves upon PEGASUS by about 5%) and almost all research consists of fine-tuning on the specific dataset or within the domain of interest.

Other research has shown various state of the art Transformers applied for summarization as well (Gupta et al., 2021). In Gupta et al., the authors apply similar Transformer models, namely BART, T5, and PEGASUS, specifically to BBC news data. Contrary to the previously discussed research, they find that T5 outperforms all other models for ROUGE scores on BBC news data but note that T5 is the only model they specifically fine-tuned on the dataset (Gupta et al., 2021). However, the authors do not discuss all of the pre-trained models they used, which is notable because model size and complexity varies within the same family, especially within T5 (Raffel et al., 2019).

Therefore, prior research shows that when all models are fine-tuned on large documents and specific datasets, BigBird should slightly outperform the other model families studied (Zaheer et al., 2020). Conversely, smaller fine-tuned models should outperform larger models not fine-tuned for the specific data or domain (Gupta et al., 2021). Though these models have use-cases in summarization, easily accessible and publicly available versions and variants discussed in the papers above are not available for all model and dataset combinations, leaving researchers to fine-tune their own model or use a pre-created checkpoint. Furthermore, research regarding the performance of pre-trained models outside of the domain they were trained for is sparse with respect to summarization. Therefore, our research differs from previous approaches in that it takes a practical approach in studying how pre-trained model variants from the Transformers library perform on datasets they were not fine-tuned on and how they compare to a simple baseline. In the Methodology section, we outline and provide links to the specific open-source models used in this study.

3.2. Dataset Comparison

The three datasets we studied are commonly studied in text summarization. Most research uses at least two of the datasets studied in this paper, and researchers often use these datasets when evaluating new models (He et al., 2020; Lewis et al. 2019; Zaheer et al., 2020). For example, CTRLSum was introduced in 2020 to allow practitioners to better account for user preferences in summarization and was also evaluated on the CNN/DailyMail and BigPatent datasets (He et al., 2020). Their popularity is further evidenced by the CNN/DailyMail dataset having over 18,000 downloads, the PubMed data having over 5,000 downloads and the BigPatent dataset having over 1,200 downloads on the Hugging Face Hub alone.[7] To compare, a majority summarization datasets have under 250 downloads.

One potential drawback of these datasets is that little literature exists comparing the linguistic differences between datasets and the effects those differences have on performance. Researchers often fine-tune and evaluate a model on different datasets without thoroughly remarking or analyzing the linguistic or structural differences between datasets (Bommasani and Cardie, 2020). In one study,

[7] https://huggingface.co/datasets?task_categories=task_categories:summarization&sort=downloads

researchers compared CNN/DailyMail and PubMed articles with metrics tied to linguistics such as abstractivity and coherence, and found that PubMed was more abstractive and coherent than CNN/DailyMail data (Bommasani and Cardie, 2020). The lack of coherence in the CNN/DailyMail data is also supported by other researchers who have examined entities, and they argue that patents and scientific papers are more coherent (Sharma et al., 2019; Bommasani and Cardie, 2020). Furthermore, the similarity between the articles and summaries (i.e. lack of abstraction) has been remarked by others (Raffel et al., 2019; Sharma et al. 2019; Bommasani and Cardie, 2020).

This review of relevant works shows that despite the widespread use of the datasets chosen in our research, some may be better-suited for abstractive summarization than others. Due to our interest in using at least three distinct domains, we chose to use all three datasets for our analyses. However, these studies give insight into key features in the data which could affect model results both in terms of quantitative error metrics and in human readability.

3.3. Evaluation Metrics

Evaluation metrics for summarization tasks have seen little progress in recent decades. 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 evaluation metrics that are more scalable for large datasets. BLEU (Bilingual Evaluation Understudy Score) was proposed in 2002 as a precision-based metric for evaluating the quality of generated texts (Papineni et al., 2002). In 2004, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) was introduced as a simpler 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. Notably, both word-overlap methods ROUGE and BLEU have their own limitations, such as not being able to sufficiently capture how much a summary resembles one written by a human (Ganesan 2018; Tatman, 2019). However, they offer automatic evaluation methods that are far less costly and time-intensive than human evaluation.

Another metric used in summary assessment is 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 provide an approximation of a model's ability to generate clear sentences, offering another large-scale substitution for human annotation (Campagnola, 2020).

4. METHODOLOGY

In accomplishing our research objectives, we first established a three-sentence baseline which Transformer-based models will be compared against, an approach identified through our literature review (Lewis et al., 2019). We chose four Transformer models to investigate and defined evaluation metrics to measure the performance of the text summarizers. For each of these approaches, we used the documents from the test split of the dataset as input. We did not fine tune the models for each data in order to standardize the training process and because the aim of our research was to study generalizability of pre-trained models.

Although we only used the smaller test datasets in this project, the only difference in application would be runtime. Additionally, we verified that our code can be run across all splits of each dataset and achieve similar results and opted to use the test data for more rapid iteration and because its purpose is for evaluation. Lastly, we had an ethical interest in preserving university-wide computing resources, so using the smaller datasets was preferable. All of the code created as part of this research is also publicly available[8].

[8] https://github.com/mkalmus/transformer_evaluation_for_summarization

4.1. Baseline Creation

Lead-Three is the name we gave to the baseline summarization method that returns the first three sentences of a given document, a baseline that is adopted in other summarization literature (Lewis et al., 2019) To apply this method, we simply join the first three sentences of an article with a newline character. We apply this to every article in the data and write the results to a file, which will later be compared to the ground truth summary.

4.2. Transformers (BigBird, PEGASUS, BART, T5)

We chose to further investigate four Transformer models: BigBird, PEGASUS, BART, and T5 for their high success in a wide variety of NLP tasks and specifically with regards to summarization. Where the Related Work gave background on the general model families, this section provides description on the specific model implementations accessed through the Transformers library. In Table 2 below, we provide the names and links to the pre-trained models used. We also report the disk space when loading models, which indicates complexity and the size of the vocabulary each model used during training.

Table 2: Parameters for Transformer Models

Model Family	Model Implementation Name (and Repository if Applicable)	Disk Size (GB)	Pre-trained Model Maximum Input Length (Tokens)
BigBird	google/bigbird-pegasus-large-bigpatent[9]	2.15	4096
PEGASUS	sshleifer/distill-pegasus-xsum-16-4[10]	1.38	512
BART	sshleifer/distilbart-xsum-12-1[11]	0.42	1024
T5	t5-small[12]	0.23	512

Using various classes and utilities within the Hugging Face ecosystem, we were able to standardize our methodology to work with any pre-trained model in the ecosystem that is available for summarization[13] with available weights and tokenizer as outlined below:

- **Tokenization**: Each pre-trained model requires the use of its own pre-trained tokenizer. To get the correct tokenizer for a given model, we utilize the AutoTokenizer[14] class and the .from_pretrained() method from the Transformers library which can easily generalize to any checkpoint by passing in the name of the checkpoint as a string (Wolf et al. 2019). Additionally, articles must be padded to the same length to work with most Transformer models. To do this effectively and eliminate repeated operation, we pre-tokenize the data in one batch, which ensures the max length padding is conducted properly and allows us to load in numeric tensors rather than full-text strings. Though batching could improve efficiency, some issues may arise with tokenizers padding to the maximum length of the batch rather

[9] https://huggingface.co/google/bigbird-pegasus-large-bigpatent
[10] https://huggingface.co/sshleifer/distill-pegasus-xsum-16-4
[11] https://huggingface.co/sshleifer/distilbart-xsum-12-1
[12] https://huggingface.co/t5-small
[13] https://huggingface.co/models?pipeline_tag=summarization&sort=downloads
[14] https://huggingface.co/docs/transformers/v4.18.0/en/model_doc/auto#transformers.AutoTokenizer

than the dataset, so we instead spread tokenization across processors and cache the result. Additionally, we made use of the Datasets library to load data in a lightweight format and apply row-wise and column-wise functions like tokenization across corpora.

- **Model Implementation:** Models were implemented in a similar fashion to the tokenizers by using the AutoModelForSeq2SeqLM[15] class and the .from_pretrained() method. This class allowed us to pull all the configurations and weights needed for a summarization model for any model pre-trained on sequence-to-sequence tasks like summarization. The model weights could then be applied with the pre-tokenized data, converted to tensors, summarized, and later evaluated. Both the model and its inputs were sent to a GPU to perform efficient tensor mathematics in generating the summaries. Work was conducted on an NVIDIA Tesla V100 accessed through the University of Michigan's Great Lakes cluster[16]. To ensure we did not run out of memory using one GPU, we wrote functions to batch inputs and pass batches to the models. Batch sizes of five were used to ensure the script worked with all models, though we found for all models aside from BigBird, batch sizes up to 32 could be used. Each model was allowed to create summaries up to its maximum input length and each model was run with eight beams as the beam search parameter.

Tokenization and implementation were conducted for all the models and datasets described throughout the paper. Once summaries were created, we compared them to the ground truth summaries and calculated quantitative metrics as described next.

4.3. Evaluation Metrics

Generated summaries were evaluated with ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and Perplexity (Campagnola, 2020). 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. Often, both recall and precision are calculated as shown below:

$$Recall = \frac{number\ of\ overlapping\ words}{total\ words\ in\ reference\ summary}$$

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

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 and helping us average out the trade-offs between the two metrics. The equation is below:

$$FScore = 2 * \frac{precision * recall}{precision + recall}$$

For the results, ROUGE-1 and ROUGE-L were used to explore possible differences between summaries. ROUGE-1 measures unigrams whereas ROUGE-L measures the longest matching

[15] https://huggingface.co/docs/transformers/model_doc/auto#transformers.AutoModelForSeq2SeqLM
[16] https://arc.umich.edu/greatlakes/

sequence of words. By using ROUGE-L, the longest common subsequence can be tracked and recorded (Lin, 2004).

BLEU works similarly to ROUGE, but penalizes cases where a token in the generated summary occurs more times than it does in the reference (ground truth) summary, and it penalizes summaries that are longer than the ground truth summary (Papineni et al., 2002). This aims to solve an issue with ROUGE in that both precision and recall can be "cheated" by using one word from the reference summary and repeating it enough times to equal the length of the reference. To provide a trivial example, consider the situation below:

- Reference Summary: "the fox jumps"
- Model-generated Summary: "the the the"

Though the model-generated summary makes no sense, its ROUGE-1 scores for both precision and recall would be 1 as per the previously discussed formulas. BLEU penalizes this case by modifying precision so that words are only counted the amount of times they occur in the ground truth summary, and therefore the numerator in the precision equation is capped. In the above example, since "the" only occurs once in the reference summary, the BLEU-modified precision would instead be 1/3. In practice, BLEU is often applied across multiple degrees of n-grams (i.e. for 1-grams, 2-grams, etc.) and we take the geometric mean of the modified precision scores up to the desired degree of n-grams. For readers interested in the mathematical formulation, we refer to the original BLEU paper (Papineni et al., 2002). For our purposes, we apply BLEU through Hugging Face's Metrics library[17], which implements the calculation with utilities for efficiency like batched calculation. With BLEU scores, we have another method to measure how often the words in the generated summaries appear in the true reference summaries.

Perplexity, was also used as a proxy for human readability as is common in NLP tasks (Campagnola, 2020; Kim, 2018). From each dataset's test split, a unigram model was built where each vocabulary term *v* has a probability *p* of appearing in the vocabulary. For words with low probabilities, a value of 0.01 is applied for smoothing. Then, for each generated summary *W,* Perplexity per word is calculated in the following formula (Kim, 2018):

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

The final evaluation metric was related to human evaluation, in which a random sample of 15 documents from each dataset was examined manually for readability and coherence. We used a Five Point Likert Scale[18] to evaluate the sampled summaries. The scores were then averaged for each model and dataset combination, which generated 15 aggregate scores. Using human evaluation, we were able to obtain a better sense of grammar and syntax correctness which may not be shown in the other calculated metrics.

[17] https://huggingface.co/docs/datasets/how_to_metrics

[18] The Five Point Likert Scale is a scoring system where 1 represents 'Completely Unreadable' and 5 represents 'Completely Readable' with each integer in between representing different degrees of readability.

5. RESULTS

After following the steps outlined in the Methodology section, we were able to generate ROUGE, BLEU, Perplexity, and human-readability scores for each dataset and model. In this section, we show and note some of the key results.

Table 3: Reference-Dependent Summary-Evaluation Metrics

	BigPatent			PubMed			CNN/DailyMail		
Models	**R-1**	**R-L**	**BLEU**	**R-1**	**R-L**	**BLEU**	**R-1**	**R-L**	**BLEU**
Lead-3	28.6	18.0	4.616	26.5	16.7	3.904	39.0	24.7	11.594
BigBird	31.8	22.2	5.142	22.7	15.6	2.335	16.1	11.5	1.827
BART	20.2	14.6	0.267	14.3	10.3	0.011	22.3	15.6	1.138
Pegasus	24.1	16.7	1.284	17.9	12.7	0.155	24.7	16.7	3.129
T5	10.0	8.3	0.003	5.9	5.2	0	19.4	15.9	0.601

Table 3 above shows the performance of each model on all datasets with respect to their ROUGE-1 (R-1), ROUGE-L (R-L) and BLEU scores, which compare the generated summaries to the ground truth summaries. The results show that the only situation in which the simple three-sentence baseline was outperformed was by the fine-tuned BigBird model on the patent data which it was specifically fine-tuned on. This indicates that the summaries created by the baseline more closely resembled the true summaries than the pre-trained models in almost all cases. Next, we show and discuss how the models performed in terms of Perplexity and readability.

Table 4: Reference-Independent Summary-Evaluation Metrics

	BigPatent		PubMed		CNN/DailyMail	
Models	**Perplexity**	**Readability**	**Perplexity**	**Readability**	**Perplexity**	**Readability**
Lead-3	776	4.60	1794	4.73	2684	5.00
BigBird	832	3.93	1549	3.33	inf	2.73
BART	379	3.80	375	3.80	1203	4.33
Pegasus	517	3.40	586	3.87	1398	4.53
T5	672	3.27	1945	2.47	2671	3.40

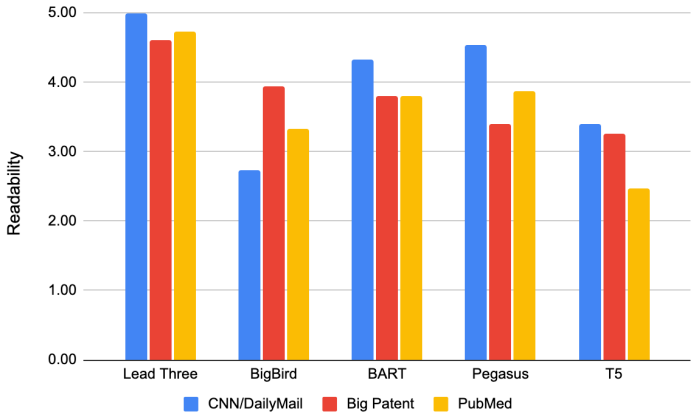


Figure 7: Readability scores of each model trained on reference datasets

Table 4 and Figure 7 above show the performance of each model on all datasets with respect to evaluation metrics that do not rely on the gold standard/ground truth. For most model and dataset combinations, the simple baseline again outperforms the Transformers for Perplexity and readability, meaning that the baseline shows less predictability with regards to what an output will be given an input. Additionally, for each dataset, Lead-3 produced the best readability scores by a wide margin. Among Transformer models, T5 produced summaries with the highest average perplexity and lowest average readability across datasets. Additionally, the three out-of-domain models outperformed Big Bird in terms of readability on the CNN/DailyMail dataset. Next, we discuss how the output summaries differed in terms of their length.

	BigPatent	**PubMed**	**CNN/DailyMail**
Lead-3	72.8%	46.5%	145.2%
BigBird	59.5%	42.3%	298.5%
BART	25.2%	13.7%	35.8%
Pegasus	35.1%	18.9%	56.8%
T5	11.2%	5.3%	22.4%

Table 5: Generated Summary Lengths in relation to Gold-Standard Lengths (i.e. 200% indicates a summary is twice as long as the ground truth)

Table 5 above shows the length of each generated summary with respect to the length of the gold standard/ground truth summaries. With the exception of the baseline applied to CNN/DailyMail, the generated summaries were all shorter, usually by a significant amount, than the ground truth summaries for both the baseline and the Transformers. In Appendix A, we include a small sample of some representative summaries from all models. Next, we evaluate model runtime in light of the performance metrics.

	BigPatent		PubMed		CNN/DailyMail	
Models	**Mins**	**Samples/s**	**Mins**	**Samples/s**	**Mins**	**Samples/s**
BigBird	174	1.37	117	0.95	170	1.00
BART	18	13.22	9	12.33	14	12.40
Pegasus	23	10.40	16	6.93	17	10.20
T5	7	34.00	6	18.49	6	28.90

Table 6: Runtime Metrics for Summary Generation by Model

Table 6 above shows the runtime of each model both with respect to the absolute summarization time in minutes, and the number of summaries generated per second. Lead-3 was not included in the table because its run time was orders of magnitude shorter than the Transformer models. For all datasets, BigBird took by far the longest to run, approximately tenfold longer than the longest non-baseline metric. This is notable because BigBird only performs slightly better than the simple baseline and other Transformers in terms of ROUGE and BLEU scores, but takes far longer to run as it is such a large model.

6. DISCUSSION

While standardization was applied where possible, differences between datasets made comparison challenging. However, we were able to draw key conclusions regarding the ability of modern Transformer architectures to generalize to data outside of the domain they were fine-tuned on for the purpose of summarization. Our observations are below:

1) **Pre-trained models without fine-tuning do not outperform a simple extractive baseline, but models fine-tuned and evaluated with the same dataset do**: While it's possible that larger distillations of the variants selected for this project could produce higher scores in ROUGE and BLEU, fine-tuning is likely to produce the best results. Given the need for fast inference speeds and lightweight models, NLP researchers and practitioners should pursue fine-tuning over larger models.

2) **The length of generated summaries significantly impacted ROUGE, BLEU, and Perplexity results irrespective of model**: The highest ROUGE scorers, Lead-3 and BigBird, generated summaries that were much longer than the smaller models. The average lengths of Lead-3 summaries were 78% of the average abstract length in BigPatent, 46.5% of the average abstract length in PubMed, and 145.2% of the average summary length in CNN/DailyMail. Because R-1 and BLEU measure overlapping *n*-grams, generated summaries with more vocabulary to choose from likely resulted in higher scores.

3) **Results were likely influenced by structural differences in datasets**: PubMed, for instance, is made up of medical journals where more important findings are likely in the conclusion. Depending on the length of the article, those findings could have been cut off given the maximum accepted input length of a sequence for smaller models is between 512 and 1024 tokens. CNN/DailyMail, a collection of short news articles, will likely offer more salient

details earlier in the piece, making summarization of those points easier. Comparing datasets with fundamentally different organizational and structural styles is a major challenge in generalizability studies such as this one. Future research should consider not just the dataset on which a pre-trained model was learned, but the necessary input sequence length for capturing all relevant information in a text.

4) **Larger models are considerably slower**: The BigBird variant used for this project was fine-tuned for BigPatent by researchers at Google, qualifying as a strong baseline for the experiments. While scores were significantly higher for all three datasets, runtime differences were notably large. For example, PEGASUS was the slowest lightweight model, but was still capable of producing 10.4 summaries per second over BigBird's 1.37 summaries per second on the BigPatent data. The tradeoffs between quality and speed should be given consideration for those interested in integrating a Transformer model into their research or application.

5) **BART emerged as a clear winner in generalizability across all three datasets**: Given ROUGE and BLEU performance was similar between BART and PEGASUS, BART's faster inference speed distinguishes itself and presents an attractive option for researchers or practitioners looking to begin their own trial and error.

7. CONCLUSIONS

In this research, we studied how well state of the art Transformer models perform for text summarization, and specifically how well they generalize to domains and data they were not trained or fine-tuned on. We apply four Transformer models with summarization use-cases: BigBird, PEGASUS, BART, and T5, as well as a simple three-sentence extractive baseline, and examine how they perform on patent data, on scientific paper data, and on news data. The results of this project proved our hypothesis wrong: pre-trained models performed worse than baselines in new domains. However, these results are in-line with other work with other models that showed smaller, fine-tuned Transformers outperform larger, pre-trained ones when evaluating on other datasets and domains. We conclude that smaller pre-trained models generalize poorly for new domains and especially when applied to completely different domains than the ones they were trained with for the models and datasets studied. Only news articles saw a boost in performance over baselines and only in some metrics, but this was likely helped by the fact that the lightweight models BART and PEGASUS were pre-trained on different news data. For NLP researchers looking for better performance while using smaller models, fine-tuning will be necessary, and more work should be conducted examining how linguistic differences in datasets affect model results and outputs.

8. OTHER THINGS WE TRIED

Throughout the course of this project, we tried a few additional techniques and methods regarding the data processing and modeling. First, due to issues with University computing issues which were later resolved, we initially wrote scripts to manually parse and store the raw datasets. Specifically, a dataset called MultiNews required scripts that were nontrivial to create to go from raw data to something that could be fed to a Transformer model. Ultimately, the MultiNews data actually did not fit our use case so we did not use the dataset. Additionally, storing and operating on the data files with pandas as we did originally was cumbersome and not easily parallelizable. Ultimately, we were able to leverage the

Datasets Python library to efficiently load and work with each of them as described in the Data and Methodology sections.

Additionally, we spent significant time writing and running code for extractive baselines that were not used in the final research. For example, we developed a method to manually extract sentences using TF-IDF, which required extensive runtime due to its need to store each token and how often it occurs in each article. We found that this did not result in sensical summaries and had extremely high computational cost. We also found through our literature review that more common approaches are to use a three-sentence baseline and compare various Transformers, and we instead adopted this approach.

Regarding modeling, we ran the models with different hyperparameters using different batch sizes and spent time analyzing how different hyperparameter settings affected the generated summaries qualitatively (i.e. looking to see if 2-grams or 3-grams were repeated many times). We experimented with different batches because some sizes caused our GPU to run out of memory with some models.

Finally, we initially conducted our work on the training data which was much larger in size (500k+ article/summary pairs across all the datasets). Doing work with the training data was useful because we verified our code can work with the full datasets, but ultimately used the test set as this research focused strictly on evaluation and the test set was curated for evaluation where the training set is largely for fine-tuning. Additionally, we found similar results comparing metrics from the training to testing sets. Since we were interested in trying different parameter combinations and the other reasons discussed above, we opted to use the test data instead.

9. WHAT WE WOULD HAVE DONE DIFFERENTLY

If we were able to take our newfound collective knowledge and copied it to our January selves, we could have spent our time much more efficiently.

The first and most significant thing that we would have done differently was to understand the true scope of our project earlier in the semester. We initially sought to create an entire Transformer model from scratch, akin to one of the ones found on Hugging Face. However, by the time we realized that it would be infeasible for us to complete with our given resources, time, and collective knowledge on hand, it was already mid March. Much of this was due to not understanding the landscape of the Transformers world, and our gradual discovery this ended up being a great learning experience.

Next, we could have introduced more baselines, both non-neural network based (such as TextRank and TF-IDF) and non-Transformer neural network based (such as LSTM and GRU). We intended to focus on some early on in the semester, but then we decided to instead prioritize learning about Transformer architecture, its applications, and which models perform best for summarization.

Additionally, it would have been interesting to see how we could have fine-tuned the pretrained models across different domains. Maybe we could have improved the performance of the models? This question is an interesting topic for future work.

Finally, we could have improved the reporting of the quality of generated summaries by using crowdsourcing to have people rate their readability, usefulness, and relevancy. The only sort of evaluation we did that came close was one group member rating a set of random samples of generated summaries and only for readability.

Overall, while we accomplished a lot in this project, it could have been enhanced further by implementing the methods as described above. Our most significant barrier towards accomplishing these was lack of time as we were limited to just a few weeks by the time we finalized the scope of the project.

10. GROUP EFFORT

Michael:

- Wrote data loading and tokenizing code
- Wrote summarization code
- Ran scripts on PubMed data

Cameron:

- Wrote Lead-3 baseline
- Wrote scripts for metric evaluation (ROUGE, BLEU, Perplexity)
- Ran scripts on BigPatent data

Harrison:

- Ran the scripts on CNN/DailyMail
- Evaluated readability scores of generated summaries

11. REFERENCES

Bommasani, Rishi, and Claire Cardie. “Intrinsic Evaluation of Summarization Datasets.” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (NLP)*, 2020. *ACL Anthology*, https://aclanthology.org/2020.emnlp-main.649/.

Campagnola, Chiara. *Perplexity in Language Models*. 18 May 2020. *Towards Data Science*, https://towardsdatascience.com/perplexity-in-language-models-87a196019a94.

Cohan, Arman, et al. “A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents.” *arXiv*, 2018. https://arxiv.org/abs/1804.05685.

Ganesan, Kavita. “ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks.” 2018. *arXiv*, https://arxiv.org/abs/1803.01937.

Goodwin, Travis R., et al. “Flight of the PEGASUS? Comparing Transformers on Few-Shot and Zero-Shot Multi-document Abstractive Summarization.” *Proceedings of COLING. International Conference on Computational Linguistics*, 2020, pp. 5640-5646.

Gupta, Anushka, et al. “Automated News Summarization Using Transformers.” *Sustainable Advanced Computing*, vol. Select Proceedings of ISAC 2021, 2021. *arXiv*, https://arxiv.org/abs/2108.01064.

He, Junxian, et al. “CTRLsum: Towards Generic Controllable Text Summarization.” 2020. *arXiv*, https://arxiv.org/abs/2012.04281.

Kim, Aerin. *Perplexity Intuition (and its derivation)*. 11 Oct 2018. *Towards Data Science*, https://towardsdatascience.com/perplexity-intuition-and-derivation-105dd481c8f3.

Lewis, Mike, et al. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.” *arXiv*, 2019. *arXiv*, https://arxiv.org/abs/1910.13461.

Lin, Chin-Yew. “ROUGE: A Package for Automatic Evaluation of Summaries.” *Text Summarization Branches Out*, 2004, pp. 74-81. https://aclanthology.org/W04-1013/.

Nallapati, Ramesh, et al. “Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond.” *The SIGNLL Conference on Computational Natural Language Learning, 2016*,

2016. *arXiv*, https://arxiv.org/abs/1602.06023.

Nenkova, Ani, and Rebecca Passonneau. “Evaluating Content Selection in Summarization: The Pyramid Method.” *Association for Computational Linguistics*, vol. Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, 2004, pp. 145-152.

Papineni, Kishore, et al. “BLEU: a method for automatic evaluation of machine translation.” *Association for Computational Linguistics*, vol. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02), 2002, pp. 311-318. https://aclanthology.org/P02-1040/.

Raffel, Colin, et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” *arXiv*, 2019. https://arxiv.org/abs/1910.10683.

Sharma, Eva, et al. “BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization.” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. *arXiv*, https://arxiv.org/abs/1906.03741.

Tatman, Rachael. *Evaluating Text Output in NLP: BLEU at your own risk*. 15 Jan 2019. *Towards Data Science*, https://towardsdatascience.com/evaluating-text-output-in-nlp-bleu-at-your-own-risk-e8609665a213.

Vaswani, Ashish, et al. “Attention Is All You Need.” *arXiv*, 2017. https://arxiv.org/abs/1706.03762.

Wolf, Thomas, et al. “HuggingFace's Transformers: State-of-the-art Natural Language Processing.” 2019. *arXiv*.

Zaheer, Manzil, et al. “Big Bird: Transformers for Longer Sequences.” *arXiv*, 2020. https://arxiv.org/abs/2007.14062.

Zhang, Jingqing, et al. “PEGASUS: Pre-training with Extracted Gap Sentences for Abstractive Summarization.” *arXiv*, 2019, https://arxiv.org/abs/1912.08777.

12. APPENDICES

12.1. Appendix A: Sample Outputs

BigPatent

Ground Truth: the method comprises the steps of providing a semiconductor body or substrate with a recess or trench in a main surface , applying a mask on the main surface , the mask covering the recess or trench , so that the walls and bottom of the recess or trench and the mask together enclose a cavity , which is filled with a gas , and forming at least one opening in the mask at a distance from the recess or trench , the distance being adapted to allow the gas to escape from the cavity via the opening when the gas pressure exceeds an external pressure .

Lead-Three: fig1 is a cross section of an intermediate product of an example of the method .
a semiconductor body or substrate 1 has a recess or trench 2 in a main surface 10 . the recess or trench 2 can be provided for a through - wafer via or through - substrate via (tsv), for instance .

BigBird: A method is described in which a mask is applied to a recess or trench in a substrate and a gas is captured in the recess or trench during the application of the mask and a pressure difference is generated between the mask and an atmosphere outside the recess or trench during further process steps.

BART: The process of creating a semiconductor that can be made for a large scale of the material that is being made available for the development of a semiconductor.

PEGASUS: A semiconductor body or substrate 1 has a recess or trench 2 in a main surface 10, and a mask 3, which is filled with ambient gas when the gas is being applied to a substrate 10, in the same way as it is to be used in a subsequent process.

T5: fig1 is a cross section of an intermediate product of an example of the method

PubMed

Ground Truth: granular cell tumor (gct) is uncommonly presented with cutaneous ulcer . we examined the clinicopathological and immunohistochemical features of this ulcerative form in fourteen cases that may raise the awareness of this variant . the study included 11 males and 3 females with a mean age 31.5 7.42 years . all cases were presented with large solitary ulcer with indurated base , elevated border , skin colored margin , and necrotic floor . twelve lesions were located on the extremities and two lesions on the genital region . histologically , the lesions showed dermal infiltrate composed of large polygonal cells with granular cytoplasm and characteristic infiltration of the dermal muscles in all cases . immunostaining showed positive reaction for s100 (14/14) , nse (14/14) , cd68 (5/14) , and vimentin (7/14) while hmb45 , ck , ema , and desmin were negative . we hope that this paper increases the awareness of ulcerative gct and consider it in the differential diagnosis of ulcerative lesions .

Lead-Three: granular cell tumor (gct) is an uncommon condition of the skin that was described firstly by weber in 1854 and established as a clinical entity by abrikossoff in 1926 who termed it as granular cell myoblastoma . the tumor occurs frequently among women and blacks , between the

second and sixth decades of life . the common location of gct is the oral cavity , but it can also occur at any other sites .

BigBird: In this study, we highlight the clinicopathological and immunohistochemical features of an ulcerative variant of granular cell tumor that is common among women and blacks in al - azhar university hospitals and the national cancer institute, cairo, egypt during the period from 2000 to 2010.

BART: The first case of a rare form of skin cancer in the 1970s and 20s has been described as as a clinical study by the University of Egypt.

PEGASUS: A study of a common ulcerative ulcer in Egypt has highlighted the clinicopathological and immunohistochemical features of a different variant of the lesion.

T5: (gct) is an uncommon condition of the skin that was described firstly

CNN/DailyMail

Gold Standard: A howler from Reading goalkeeper Adam Federici helped Arsenal progress . Federici let a late effort squirm through his legs and in during extra-time . He left the field in tears after fumbling Sanchez's shot through his legs . Aaron Ramsey will be hoping to taste more FA Cup success in the final .

Lead-Three: Aaron Ramsey was relieved after Arsenal ground their way past Reading in extra time to return to the FA Cup final. Alexis Sanchez's two goals, the second via a howler from Reading goalkeeper Adam Federici, saw the defending champions through despite Garath McCleary's goal for the Royals. Ramsey told BBC One: 'We knew it was going to be a tough game, they showed great character to come back into it and we were on the ropes at times.

BigBird: A method whereby a first team competes with a second team for the right to advance to the final of a pre-determined competition. The first team competes against the second team for the right to advance to the final of the pre-determined competition, while the second team competes against the first team for the right to advance to the final of the pre-determined competition. The first team competes against the second team for the right to advance to the final of the pre-determined competition, while the second team does not compete against the first team for the right to advance to the final of the pre-determined competition, but rather competes against the first team for the right to advance to the final of the pre-determined competition in order to advance to the final of the pre-determined competition in order to advance to the final of the pre-determined competition in order to advance to the final of the pre-determined competition in order to advance to the final of the pre-determined competition in order to advance to the final of the pre-determined competition in order to advance to the final of the pre-determined competition in order to advance to the final of the pre-determined competition in order to advance to the final of the pre-determined competition in order to advance to the final of the pre-determined competition in order to advance to the final of

BART: Arsenal midfielder Aaron Ramsey says his side were "on the ropes" after beating Reading in the FA Cup final at Wembley.

PEGASUS: Arsenal are through to the FA Cup final for the second year in a row after a dramatic

extra-time win over Reading at Wembley, according to the club's captain.

T5: fought their way past Reading in extra time to reach the final. Alexis Sanche