GenAI based Text Summarization using WikiHow benchmark

Muhammad Haroon
Michigan State University
haroonmu@msu.edu

Tashfain Ahmed
Michigan State University
ahmedt11@msu.edu

Javen Zamojcin
Michigan State University
zamojci1@msu.edu

Michael Ewnetu
Michigan State University
ewnetumi@msu.edu

***Abstract*—WikiHow dataset consists of textual articles based on diverse topics which makes it one of the suitable choices for text summarisation-based research benchmark. However, top transformer models like T5, BART, etc are not evaluated on this benchmark which creates uncertainty regarding their performance on diverse topics-based text from summarisation standpoint. In this research article, we will study different model architectures for the task of textual summarisation using the WikiHow dataset and share our findings.**

I. Introduction

Along with the growth of the internet and big data, many individuals are overwhelmed by the vast amount of information and documents available online. This influx of data triggers the need for technological approaches that can automatically summarize texts, enabling users to quickly grasp essential information without losing the original intent of the documents. Text summarization research, which began in the mid-20th century, has evolved significantly, employing statistical techniques like word frequency diagrams initially and advancing to more sophisticated methods. Automatic text summarization now generates summaries that encapsulate important sentences and relevant information from the original text. The exponential daily growth of web resources—such as websites, user reviews, news, blogs, social media, and extensive archives like news articles, novels, books, and scientific papers—makes it increasingly challenging for users to locate and comprehend information efficiently. Many texts contain repetitive or irrelevant content, heightening the urgency for effective summarization methods. Manual summarization is not only time-consuming and costly but also impractical on a large scale. Automatic Text Summarization (ATS) systems address this issue by distilling the main ideas of a document into a condensed form, minimizing repetition, and saving users significant time and effort. These systems aim to deliver concise summaries that maintain the essence of the original documents, fulfilling the specific needs of various users and tasks. Text summarization is useful in different aspects, including educational purpose (aiding patients with learning difficulties) news summary, extracting actionable insights from lengthy business documents/reports, and quick summarization of patient history.

Models such as T5 and GPT have shown remarkable potential in generating concise and coherent summaries of large textual data. This project utilises the WikiHow dataset to compare the effectiveness of different models, focusing on various stages of model development, including preprocessing, fine-tuning, and performance evaluation.

II. Literature Review

The advent of LLMs has significantly reshaped the realm of automatic text summarization (ATS)—establishing new benchmarks by improving the accuracy and coherence of automated summarization, outperforming previous approaches [8]. Statistical models, while having the advantage of simplicity and computational efficiency, are limited in their ability to extract grammatical and contextual relationships—this is due to their designed reliance on word frequency rather than constructing the underlying understanding of the semantics of words and documents. Word embedding models, such as Word2Vec and GloVe, are proficient in capturing the word representations and learning specific text patterns, but are still limited in their ability to capture deeper text semantics due to their relatively shallow network architecture, especially when compared with transformer-based architectures [10]. The Transformer-based encoder used in BERT [22] has demonstrated a significant improvement in capturing deep contextual semantic information, while Transformer-based decoder only architectures, such as GPT, have demonstrated robust summarization capabilities [10]. Transformer Encoder-Decoder based models such as Pegasus [14] and BART [1], prior to the emergence of LLMs served as the cornerstone of generative text summarization.

Koupaee et al [9], evaluated the aforementioned WikiHow dataset using the architectures TextRank, Seq-to-seq with attention, Pointer-generator, Pointer-generator + coverage, with Lead-3 as the baseline, but did not evaluate any LLM-based approaches. MatchSum (BERT-Base) [10] and BertSum [11] are BERT-based approaches which have since scored higher evaluation metrics (ROUGE-1) [12] than the original study. No studies have been performed yet on this dataset using bigger-than-BERT models.

Pegasus [14] achieved state-of-the-art summarisation results on WikiHow benchmark however, it's mainly trained on English language which makes it uncertain for multilingual text summarisation while T5 model is tested for summarisation in 101 languages which makes it a better candidate for standard text summarisation model. The mT5-small model has shown impressive versatility in language processing tasks across different tasks. In one study [25] , it tackled the challenge

of abstractive summarization of dialogues in three widely spoken Indian languages: Hindi, Marathi, and Bengali. The researchers assessed how effectively the mT5-small model could handle the nuances and complexities of these languages in a dialogue context.

Another area [26] of exploration involved adapting the mT5-small model for different pretraining configurations. This included testing variations in pretraining data quality, optimization strategies, and the duration of pretraining, with a particular focus on the Portuguese language. Furthermore, the model was compared with its larger counterpart [27], the mT5-base, in summarizing Persian news texts. The flexibility of the mT5 model was also highlighted in [28] with applications such as Sentiment Analysis, Question Generation, and Question Answering, confirming its broad utility in specialized NLP tasks.

Moreover, alternative approaches to using larger pre-trained encoder-decoder models, such as T5 and Pegasus, for sequence-to-sequence tasks have been actively explored. Rothe et al. [8] introduced a framework that bypasses the computationally expensive pre-training of encoder-decoder models by leveraging pre-trained checkpoints from existing language models, such as BERT (for the encoder) and GPT-2 (for the decoder). This warm-starting technique allows for the initialization of either the encoder, the decoder, or both with pre-trained weights, enabling faster convergence during fine-tuning while retaining strong downstream task performance. By eliminating the need to pre-train the entire encoder-decoder architecture from scratch, the authors demonstrated that warm-started models can achieve competitive or even comparable results to fully pre-trained models like T5 and Pegasus, all while significantly reducing computational costs

Additionally, weight sharing has emerged as a key optimization technique to improve efficiency in encoder-decoder models. Raffel et al [23]. demonstrate the efficiency of weight sharing in encoder-decoder models, showing that a randomly initialized model where the decoder's weights are shared with the encoder achieves nearly the same performance as its non-shared counterpart. This approach significantly reduces the memory footprint, effectively halving the required parameters without substantial degradation in performance. Weight sharing ensures that layers at the same positions in both the encoder and decoder, such as the query, key, and value projection matrices in the self-attention mechanism, share identical parameters. In addition to reducing the memory footprint, this technique enables warm starting an encoder-decoder model by initializing it from a pre-trained encoder-only checkpoint, such as BERT. For this method to work, the decoder's architecture (excluding cross-attention layers) must be identical to the encoder's structure. This architectural symmetry ensures that weights can be seamlessly shared. By reusing the encoder weights in the decoder, warm-started models benefit from the transfer learning capabilities of pre-trained encoders, achieving competitive performance at a fraction of the training cost associated with fully pre-training large encoder-decoder models like T5 or Pegasus. This makes the approach particularly appealing for tasks requiring memory efficiency and computational scalability [24].

III. About Dataset:

The WikiHow dataset is a large-scale resource designed for text summarization tasks, offering over 200,000 article-summary pairs derived from the online WikiHow knowledge base. Unlike traditional datasets, such as CNN/Daily Mail, Gigaword, or DUC, which primarily focus on news articles with specific writing styles and limited abstraction levels, the WikiHow dataset covers a wide range of topics and diverse writing styles [11]. Each article in the dataset is constructed by merging multiple paragraphs, while summaries are created by combining paragraph outlines, providing a structured format that supports long-sequence abstractive summarization. This dataset addresses key challenges in summarization, including the need for high diversity, large-scale data, and support for abstractive systems that generate summaries at deeper semantic levels. Koupaee et al highlight the dataset's novelty through metrics like abstraction levels and compression ratios, demonstrating its suitability for abstractive summarization tasks. Additionally, benchmarks on the dataset with extractive and abstractive models emphasize its complexity and usefulness for advancing summarization research. By overcoming the limitations of existing datasets, WikiHow serves as a valuable resource for training and evaluating generalized sequence-to-sequence models in text summarization [11].

IV. Model Architectures

A. T5 Model

T5 (Text-to-Text Transfer Transformer) [13] is an encoder-decoder model designed to handle natural language processing (NLP) problems as text-to-text tasks. T5-base version consists of approximately 220 million parameters, establishing a good tradeoff between performance and computational efficiency. Pretraining approach is based on Large-scale unsupervised training using "Colossal Clean Crawled Corpus" (C4 dataset). T5 is based on unified framework where both the input and output are processed as text strings. The model consists of an encoder-decoder structure, where the encoder processes the input sequence to produce contextualized representation and the decoder generates the output sequence in an autoregressive manner using cross-attention with encoder's hidden state, predicting one token at a time. During pretraining, T5 is trained on a span-corruption objective: parts of the input text are replaced with special sentinel tokens and model learns to predict missing spans. This denoising approach ensures that the model captures both local and global associations. T5 uses relative positional embeddings and incorporates layer normalization for stability and efficiency. It also supports scaling across sizes, from small (T5-Small) to very large (T5-XXL), enabling adaptability to computational constraints and dataset sizes. By unifying task formats and leveraging transfer learning, T5 achieves close to state-of-the-art performance on many NLP benchmarks.

B. DistilBART

DistilBART [3] is a distilled version of the BART (Bidirectional and Auto-Regressive Transformer) model [1], optimized specifically for text summarization tasks. Using a similar knowledge distillation method to DistilBERT [4], DistilBART captures the competitive performance of the original BART architecture while achieving a significantly smaller model size and faster inference times [5]. This particular DistilBART instance ("distilbart-xsum-12-3") was pretrained on the XSUM benchmark [6], and contains approximately 255 million parameters.

C. mt5-small

The mT5-small model [17] was chosen to summarize WikiHow articles primarily to investigate the trade-off between computational efficiency and the performance of smaller transformer models, as well as to explore patterns in multilingual datasets that transcend language boundaries—a research question previously discussed [19]. The hypothesis was that despite its reduced size, mT5-small could deliver competitive summarization quality due to its robust pre-training on diverse datasets.

This experiment examines the capabilities of mT5-small in terms of operational efficiency, including training and inference times. The mT5 model, pre-trained on a massive multilingual dataset covering 101 languages [17], consists of an encoder-decoder architecture built with multiple transformer layers, each comprising self-attention and feed-forward sublayers. The encoder processes the input text, encoding tokens into contextualized embeddings that capture the nuances of the input language. The decoder, optimized for sequence generation tasks, employs self-attention mechanisms for its generated output and cross-attention layers to integrate representations from the encoder. This design enables the model to attend to relevant parts of the input sequence when generating each token in the output. A language modeling head atop the decoder predicts the next tokens in the sequence. Although smaller models like mT5-small may raise concerns regarding their performance due to a reduced parameter count, prior research [18] has demonstrated that mT5-small achieves competitive results in text processing tasks, making it a promising candidate for this investigation.

D. Bert2Bert

In order to leverage publicly available pre-trained models and develop a Transformer-based sequence-to-sequence model, an architecture with BERT as both encoder and decoder was built [7]. The encoder is used to process the input text, while the decoder is used to generate output sequences. The encoder uses pre-trained bidirectional attention to encode input tokens into contextualized embeddings through 12 transformer layers, each with self-attention and feed-forward sublayers. The decoder, modified for sequence generation tasks, incorporates both self-attention for its input and cross-attention to integrate the encoder's representations. A language modeling head atop the decoder predicts the output tokens. Even though it is argued that the pre-training objective used by BERT is not well suited for tasks that require decoding texts, similar work by Rothe et al showed great results, hence, it promoted this particular work [8].

V. Methodology

This section summarizes different experimented approaches to fine-tune models for the WikiHow dataset.

A. T5

Preprocessing steps include adding the "summarize: " prefix to input texts (T5 model-specific formatting); tokenizing inputs with maximum length of 512 tokens and targets with maximum length of 128 tokens which are chosen by analysis of WikiHow dataset. This token length configuration covers the majority of text in the dataset as shown in Figure-1 and Figure-2. Furthermore, this configuration also achieves an input text to label text ratio of 0.25 which covers the majority of data as evident in Fig-3. Preprocessing also includes replacing padding tokens in labels with -100 to ignore them during training; utilizing splitted train(93.3%), validation(3.3%), and test set(3.3%). Furthermore, T5 uses SentencePiece [20] to encode text as WordPiece [20; 21] tokens using vocabulary of 32,000 word pieces.

Training strategy is based on utilizing T5-base pre-trained model for text summarisation using sequence to sequence modelling approach with early stopping and optimal checkpoint saving using ROUGE-1. Hyperparameters are Learning rate = 3e-5, Batch size = 32, Weight decay= 0.01, Number of training epochs= 10 with mixed precision training enabled. Evaluation method is based on ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and METEOR (Metric for Evaluation of Translation with Explicit ORdering).

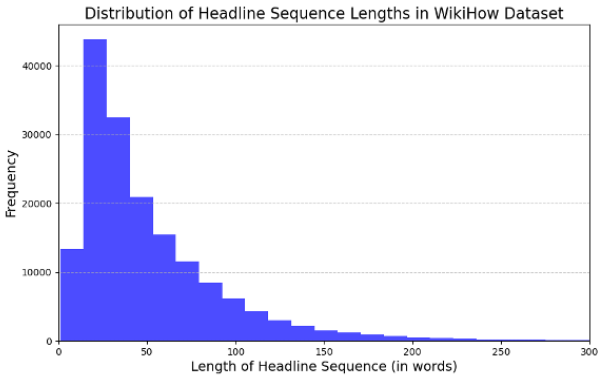

Fig. 1. Frequency distribution of label lengths.

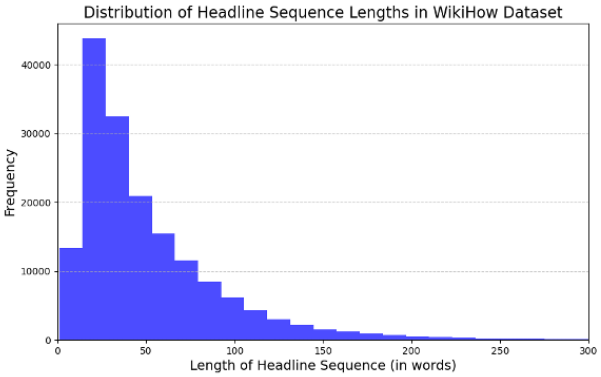

Fig. 2. Frequency distribution of input text lengths.

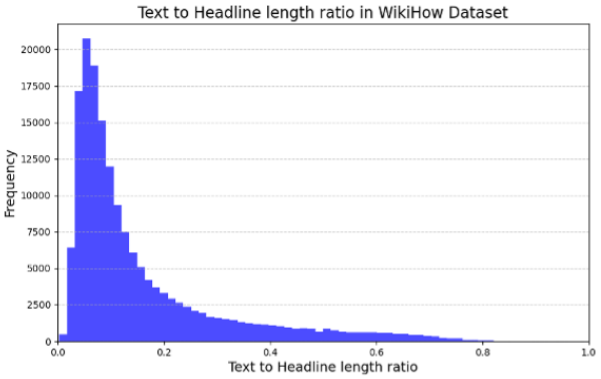

Fig. 3. Text length to headline(label) length ratio.

B. Bert2Bert

The Bert2Bert model was fine-tuned on a cleaned Wikihow Cleaned Dataset, which contained about 214 thousand samples (401 null entries), for the task of text summarization. Preprocessing involved converting all text to lowercase to align with the uncased version of the model. Then, the preprocess continued on the text and summarization columns, ensuring consistent text formatting. All punctuations were removed except periods (.), commas (,), and hyphens (-) and spaces were inserted around these retained punctuation marks. Additionally, single spaces replaced multiple spaces and eliminated leading or trailing whitespace and newline characters.

Then, the input text and summaries were tokenized with padding to ensure uniform sequence lengths and truncation to specified encodermaxlength and decodermaxlength values, respectively. Fields such as inputids and attentionmask for the encoder and decoderinputids and decoderattentionmask for the decoder were added to prepare the data. For training, the decoderinputids were copied to the labels field, with padding tokens replaced by -100 to exclude them from the loss computation.

The pre-trained BERT model served as both the encoder and decoder in a Seq2Seq fine-tuning setup. Training was performed with a batch size of 32 over two epochs, employing a gradient accumulation step of 1 and mixed precision (fp16) for efficiency. Checkpoints were saved every 3,000 steps, with only the best model (based on ROUGE-1) retained during training processes involving evaluation. Evaluation metrics included ROUGE, METEOR, and BLEU, which were calculated on a subset of the validation dataset using a custom evaluation function. Predictions were generated with beam search, constrained maximum lengths, and early stopping. Hyperparameters like a warmup of 200 steps, pinning memory, using multiple data loader workers, and grouping sequences by length optimized training.

C. mT5-small

A cleaned version of the dataset is used, involving several preprocessing steps. All words are converted to lowercase, and all punctuation is removed except for ".", ",", and "-". Spaces are added before and after all punctuation. NA values are dropped from the dataset, and leading/trailing newline and space characters are removed. These changes facilitate easier tokenization. Additionally, columns unnecessary for the summarization task, such as title, are removed to focus on the main text and its corresponding summary.

The dataset is initially filtered based on summary length to create a balanced set where summaries are close to a target median length, facilitating training around the desired length (128). A weighted sampling method is applied, where entries nearer to the desired median summary length are more likely to be included. This method emphasizes typical cases and smoothens the distribution around the median, which is plotted to visualize the effect of this sampling.

The dataset is split into training, validation, and test sets using predefined ratios, ensuring the model is evaluated on unseen data. The text and summaries are tokenized using the mT5 tokenizer, which converts the text into input IDs that the model can process. Tokenization includes truncation to manage texts exceeding the model's maximum input length. Training arguments are defined using Seq2SeqTrainingArguments, specifying parameters such as the output directory, batch size, number of epochs, evaluation strategy, and other settings crucial for effective learning and logging. A Seq2SeqTrainer is configured with these arguments, the model, tokenized datasets, a data collator (to handle data padding to uniform lengths), and a function to compute metrics during evaluation.

A similar experiment was conducted with a target output length of 30, 8 epochs, and no data balancing to test the model's robustness in a title generation task.

Both experiments involved checkpoints, a learning rate of 5.6e-5, a weight decay of 0.01, and specific epoch counts: 4 for summary generation and 8 for title generation. Finally, the model's performance is evaluated using the ROUGE metric.

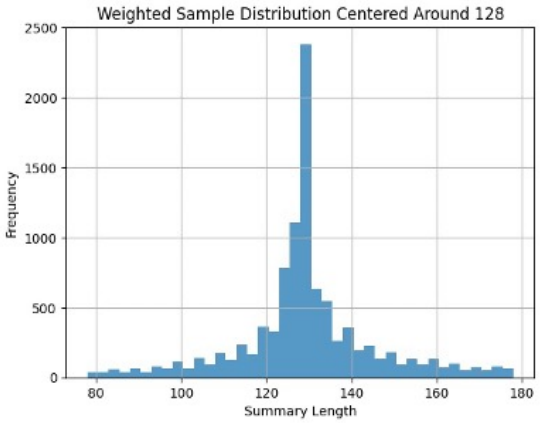

Fig. 4. Frequency distribution of label lengths after pre-processing.

D. DistilBART

In regards to preprocessing, we are using the WikiHow cleaned dataset [7]: all words are made lowercase; null entries are dropped; all punctuation removed except for periods, commas and hyphens; spaces are added before and after all punctuation; leading and trailing newline and space characters are removed; the dataset is split into training, validation, and testing sets (75.0/12.5/12.5); input text documents are truncated to 256 tokens; input summary documents are truncated to 128 tokens; input sequences are padded to multiples of 8; and finally padding tokens (-100) in labels are replaced with `<pad>` to ignore them during training.

Regarding training strategy, the pretrained model used in training is "distilbart-xsum-12-3" [3], and the task being text summarization (converting full text to headlines). The training approach was fine-tuning the pre-trained DistilBART model on WikiHow dataset using sequence to sequence modelling approach and the best model is picked at the end of training based on ROUGE-1 score. For the hyperparameters used: a learning rate of 0.005; a batch size of 32 (training) and 32 (evaluation); a weight decay of 0.01; a number of training epochs of 4; and with BF16 (mixed precision) training enabled.

The evaluation process involved computing metrics on decoded predictions, and labels and replacing default padding tokens and skipping them in the decoding process. The two metrics calculated were ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and METEOR (Metric for Evaluation of Translation with Explicit ORdering).

VI. Evaluation methods & Results

To evaluate the effectiveness of each summarization model, these metrics are used: ROUGE (Recall-Oriented Understudy for Gisting Evaluation, R1, R2, RL [5] and METEORscore, and Human Evaluation.

TABLE I
Performance metrics of various models on ROUGE and METEOR scores

Model	Rouge-1	Rouge-2	Rouge-L	Meteor
T5 (uncleaned dataset)	0.22	0.07	0.18	0.2
DistilBART (cleaned dataset)	0.33	0.0	0.22	0.066
Bert2Bert (cleaned dataset)	0.2884	0.0875	0.2059	0.2425
mT5-small (cleaned dataset)	0.2459	0.0738	0.2145	0.1697

A. Test Sample-1

Original Text (first 500 chars): Caged animals such as hamsters, guinea pigs, rodent, reptiles, and amphibians can be taken to a friend's or sitter's home. Create a document that outlines the feeding and water needs, cleaning schedule, and temperature control. Pack all of the things that mimic your pet's environment at your home such as bedding, heated surfaces, and decorations.If the cage is not mobile, someone will need to come check on your pet daily. A rabbit, ferret, or guinea pig is live bait in the wild. Relocating your

T5 Generated Summary: Make a list of your pet's needs. Find a place that mimics your pet's environment. Find a sitter. Choose a kennel.

Bert2Bert Generated Summary: create a home for your pet . move your pet to a new home in your home . keep your pet in a cage . feed your pet a healthy diet . provide your pet with plenty of exercise . provide a variety of toys and chewable toys . provide plenty of playtime and playtime

m-T5-small Generated Summary: take your pet to a friend orsitters home. collect all of the items you need for feeding and water careful things that are commonly affected by caged animal activities

B. Test Sample-2

Original Text (first 500 chars): Every boarding barn has people who go on vacation, or people who'd love to take a break from mucking stalls every day. Or even just riding all the time. Set up a rate schedule, depending on how much care you are willing to do. Not only will you get a little bit of income, but you'll learn a lot about horse-ownership along the way.; , Offer to pull manes, braid, polish hooves, etc. If you're boarding your horse in a show barn, you can probably find customers fairly easily. Braiding, especially, c

T5 Generated Summary: Set up a boarding rate schedule. Find customers. Find a good babysitter. Ask if you could school horses for them. , Ask for a coach.

Bert2Bert Generated Summary: make a list of the people you will be boarding . set up a schedule for your boarding . purchase a horse . buy a horse from a reputable breeder . get your horse used to the horses . get the horses used to your horses . make sure your horse is well groomed and groomed

m-T5-small Generated Summary: take a break from mucking stalls every day. learn about horse-ownership at the same time, if you want to do it with your friends and family

C. Test Sample-3

Original Text (first 500 chars): Let him know that you like him by making an effort to look nice whenever you're around him. You should still be yourself, but take extra care with your hair and makeup and outfits, so he can start to notice you. You don't have to wear a tight dress and high heels if you're at a baseball game with him, but let him know that you care about your looks when you're around him.. Don't be afraid to be a little sexy. If you're comfortable with your body, show it off. If you're not comfortable with a l.

T5 Generated Summary: Look nice. Make eye contact. Flirt with him.

Bert2Bert Generated Summary: be nice to him . be confident in your appearance . dont be afraid to talk to him in front of other people . be a good listener . make sure youre comfortable with your body language . be aware of the signs of a l . i . c . a . get to know his friends .

m-T5-small Generated Summary: look nice whenever you are around him. take extra care with your hair and makeup instead of wearing high heels at the same time as possible,

D. Test Sample-4

It can be hard to tell if someone is interested, but you should be able to tell if he makes an effort to spend time with you. If he initiates conversation and smiles a lot, then there's a good chance that he is interested. That said: many guys get shy when they like someone, and you shouldn't write someone off just because he hasn't made a move. This can be anything from a trip to the beach to a house party. Some guys like to make the first move, but most will respect a girl who goes for wha.

T5 Generated Summary: Find out if he is interested in you. Get to know him before you ask him out. Be yourself.

Bert2Bert Generated Summary: pay attention to how he talks to you . ask him if hes interested in making a move . make sure hes willing to make a move on you . dont be afraid to ask him out on a date . be prepared for rejection . be ready to wait for him to move on .

m-T5-small Generated Summary: tell if he is interested . write off the conversation and smiles instead of flirting with him, but you shouldn be afraid to ask for help

VII. Results Analysis

A. T5

ROUGE-1 (0.22), Measures unigram (single word) overlap between generated and reference summaries. Score of 0.22 indicates moderate content similarity and suggests the model captures about 22% of individual words correctly. ROUGE-2 (0.07), Measures bigram (two-word phrase) overlap. Lower score (0.07) suggests challenges in preserving exact word sequences. Indicates some difficulty in maintaining precise phrase-level coherence. ROUGE-L (0.18), Measures longest common subsequence between generated and reference summaries. Score of 0.18 suggests moderate structural similarity. Indicates the model maintains some overall sentence structure.

Meteor (0.2), Measures semantic similarity by performing semantic matching using WordNet. Considers synonyms and paraphrasing. Score of 0.2 suggests moderate semantic alignment. Indicates reasonable semantic understanding.

B. Bert2Bert

The model demonstrates promising yet moderate performance across all the evaluation metrics. The results suggest a balanced approach to summary generation, with ROUGE 1, unigram preservation, showing the strongest correlation at 0.2884, while capturing precise multi-word phrases proves more challenging, as evidenced by the lower ROUGE 2, bigram, and BLEU, translation-oriented scores. The model appears to maintain a reasonable semantic structure, with METEOR indicating some success in capturing underlying meaning beyond exact word matches. However, the relatively constrained metric values across ROUGE, BLEU, and METEOR collectively point to opportunities for refinement, potentially through advanced training techniques and architectural adjustments to enhance the model's summarization capabilities.

C. mT5-small:

The mT5-small model's performance on text summarization, as measured by the ROUGE metric, shows moderate capability with scores of 24.59% for ROUGE-1, 7.38% for ROUGE-2, and 21.45% for ROUGE-L. The results show that the model can capture the most frequent words and some grammatical structures. Given its multilingual training, the mT5 can theoretically handle summarizations across different languages. However, it struggles with more complex semantic relationships and maintaining contextual coherence.

TABLE II
Performance metrics of mT5-small model 30 output Len on ROUGE scores

Model	Rouge-1	Rouge-2	Rouge-L
mT5-small (cleaned)	0.28411	0.1227	0.25167

D. DistilBART:

The DistilBART model on text summarization using ROUGE and METEOR metrics produced interesting results. The ROUGE-1 score of 33.0% was the highest of the four model experimented on in this work, and higher than the SOTA MatchSUM score of 31.85% [14], but still lower than the highest SOTA score of 35.91% by the BertSUM model. This strong performance for the ROUGE-1 metric indicates DistilBART is excellent for the unigram preservation text summarization task. However, DistilBART performed exceptionally poorly on the ROUGE-2 metric, the lowest of the four models, with a score of 0.0%. I'm not sure if this is due to the limitations of this particular model, or if an implementation error is a factor in this. The ROUGE-L metric also had the highest performance of the four models with a score of 22.0%, but another poor score of 6.6% for the METEOR metric.

VIII. Conclusion

Model Performance: The results demonstrate that the choice of pre-trained model and dataset preprocessing significantly impacts performance on abstractive summarization tasks. DistilBART, Bert2Bert, and mT5-small outperform T5 when trained on a cleaned dataset, suggesting that data preprocessing plays a crucial role in improving model generalizability and coherence.

Dataset Quality Matters: Models trained on cleaned datasets (e.g., DistilBART, Bert2Bert, and mT5-small) achieved higher Rouge-L scores than T5 on the uncleaned dataset. This underscores the importance of robust data preprocessing for effective summarization.

Contextual Understanding: Bert2Bert consistently generates summaries with better contextual alignment, as reflected in its superior Rouge-2 and Meteor scores, particularly capturing nuanced relationships between entities mentioned in the text.

Trade-offs in Simplicity vs. Completeness: While T5 often produces concise summaries, it tends to omit critical details, reflected in lower Rouge-1 score. On the other hand, Bert2Bert and mT5-small provide richer detail, albeit sometimes at the cost of brevity.

Model Variations in Summarization Style:

T5: Prioritizes brevity and high-level instructions but often lacks detail and coherence. DistilBART: Generates slightly fragmented summaries, possibly due to challenges in coherence, as seen in lower Meteor scores. Bert2Bert: Strikes a balance between detail and readability, producing more coherent summaries with higher Meteor and Rouge-2 scores. mT5-small: Focuses on detail and captures nuanced context well but can exhibit redundancy or stylistic quirks. Metric-Based Insights: Rouge-1 and Rouge-L scores indicate the models' ability to preserve salient information. Rouge-2 shows that Bert2Bert performs best in capturing bi-gram level details, critical for nuanced understanding. Meteor highlights the semantic alignment between summaries and reference texts, with Bert2Bert showing the best semantic relevance. Recommendations: Based on the findings, Bert2Bert emerges as the most balanced model for summarization tasks on cleaned datasets, offering a good trade-off between brevity and comprehensiveness. Future work could explore fine-tuning on task-specific datasets and leveraging advanced preprocessing to enhance performance further.

IX. Future Work

Nature of topic is of vital importance for text summarisation, a model achieving state-of-the-art summarisation scores for history based dataset might not perform well for sports highlight based data. One approach which is worth further experimentation is to train multiple encoder-decoder models for different topics and create an ensemble of all trained models to generate final summarisation. We can train additional linear layers to pick which model summary should have higher weightage based on embedding of input text.

For the Bert2Bert model, initial fine-tuning was performed on a version of the Wikihow dataset that was not specifically designed for text summarization tasks. The dataset had short phrases as summary that look more like titles. Hence, the model achieved relatively low scores across all metrics. However, when the dataset was changed to a version with appropriate summary length, the model showed significant improvement across all metrics. Hence, this promotes the idea that using a more comprehensive dataset, possibly with more samples, like PigPatent, may lead to better performance.

The DistilBART model struggled heavily with the ROUGE-2 and METEOR metrics. ROUGE-2 refers to the overlap of bigrams between the system and reference summaries. It is interesting that the unigram overlap score (ROUGE-1) and longest common subsequence score (ROUGE-L) would be good, but not for bigram overlap. From our understanding, ROUGE scores can be sensitive to the choice of preprocessing techniques used during feature extraction, such as adjusting the n-grams length or the removal of stop words. In the future work, we would like to experiment with the preprocessing steps for DistilBART to see if the ROUGE-2 score may be improved.

For the mT5-small model, the next step should focus on generalizing its performance from imperative sentences, which are biased towards the dataset, to a broader summary generation task. This may involve diversifying the dataset by integrating other datasets to mitigate existing biases. Additionally, alternative pre-processing steps should be explored, prioritizing semantic-based filtering rather than relying solely on quantity. These steps could include more advanced filtering methods, such as labeling dataset summaries based on their quality. Implementing such improvements may require developing or utilizing existing NLP tools to assess the relevance and informativeness of summaries. For instance, machine learning models could be employed to predict quality scores for summaries based on factors such as coherence, conciseness, and coverage of critical points.

References

[1] Jin, H., Zhang, Y., Meng, D., Wang, J., Tan, J. (2024, March 5). A comprehensive survey on process-oriented automatic text summarization with exploration of LLM-based methods. arXiv.org. https://arxiv.org/abs/2403.02901

[2] MatchSum https://paperswithcode.com/paper/extractive-summarization-as-text-matching

[3] BERTSum https://paperswithcode.com/paper/abstractive-summarization-of-spoken

[4] https://paperswithcode.com/sota/text-summarization-on-wikihow

[5] Ba, L. J., Kiros, J. R., Hinton, G. E. (2016). Layer Normalisation. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition.

[6] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting.

[7] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.
[8] Goyal, T., Azizi, M., Abbeel, P. (2023). Direct Preference Optimization. Chen, C., Yin, Y., Shang, L., et al. (2022). bert2BERT: Towards Reusable Pretrained Language Models.
[9] Huggingface. (n.d.). WikiHow-cleaned Dataset.
[10] Keras. (n.d.). AdamW Optimizer Documentation.
[11] Von Platen, P. (n.d.). BERT2BERT for CNN Dailymail [Jupyter notebook]. Google Colab.
[12] https://fabianofalcao.medium.com/metrics-for-evaluating-summarization-of-texts-performed-by-transformers-how-to-evaluate-the-b3ce68a309c3