# Quality Evaluation of Summarization Models for Patent Documents

Junhua Ding[1]*, Haihua Chen[1], Sai Kolapudi[1], Lavanya Pobbathi[1], and Huyen Nguyen[1]
[1]Dept. of Information Science, Denton, TX, USA
{junhua.ding, haihua.chen}@unt.edu, {saikolapudi, LavanyaPobbathi, huyennguyen5}@my.unt.edu
*corresponding author

*Abstract*—Several recently developed neural network models have shown their potentials on automated text summarization. However, the evaluation results of these models on summarization of long text are fairly close in almost every major evaluation parameter. None of these models including large language models GPT-3.5 and GPT-4 can well summarize long text without manual interventions. In this paper, we report the evaluation results of several state-of-the-art neural network models on text summarization under different configurations of the input, which includes 1630 U.S. patent documents. Based on the evaluation results, we proposed a strategy for improving the text summarization in long text and demonstrated its effectiveness with new cases.

*Keywords–text summarization; large language model; quality evaluation; patent document*

## 1. Introduction

Text summarization is the process of producing a concise and coherent summary while preserving key information and meaning for a longer text [1]. Since text summarization enables users to quickly grasp key points of lengthy texts and efficiently access relevant information, it has been applied widely in various domains such as scientific, medical, and legal documents. For example, over 5.14 million academic articles, including short surveys, reviews, and conference proceedings, were published in 2022 [2]. Automated text summarization is an ideal way for quickly understanding these articles.

There are two major approaches of automatic text summarization: extractive and abstractive. Extractive summarization is a method of summarizing text by selecting important sentences or phrases from the original document. This approach is often modeled as a sequence labeling task, where each sentence is categorized based on whether it serves as a summary sentence or not. Extractive summarization is considered to be faster, simpler and more accurate because exact sentences are excerpted from the original document. However, it is less fluent and less coherent than abstractive summarization [3]. On the other hand, the abstractive summary is not directly excerpted from the document like the extractive method. It generates the summary with sentences that are different from those in the original text but without changing the central facts and ideas. Most current abstractive models are based on neural sequence-to-sequence learning [4], [5]. With the remarkable achievements of large pretrained language models and natural language generation, recent research has shifted gears from extractive summarization to abstractive summarization. Nevertheless, the abstractive summarization still remains as a difficult task such as models suffer from hallucinations [6], [7] and the summarization does not align with human's preferences and expectations [8]. A preliminary study of text summarization techniques indicates that nearly 30% summaries generated by existing SOTA neural abstractive summarization are unfaithful to original documents [6]. Existing research results show combining extractive and abstractive methods in a hybrid approach could be a promising solution to address the hallucination issues in abstractive methods and incoherence in extractive methods. [9] leverages contextualized rewriting method to improve the abstractive summarization using the whole document as input. They consider contextualized rewriting as a seq2seq learning problem, and use language model BERT as encoders in both the extractive summarization task and the abstractive rewriters. They also utilize semantic group alignment to enhance the document representation by aligning the extractive summary and the reference in the abstractive rewriter. Recently, transformer-based neural models like BART [10] and PEGASUS [11] have achieved SOTA results on short text summarization benchmarks. But their performance significantly degrades on longer text with thousands of tokens, even the model was extended with attention mechanisms [12], [13] or hierarchical architectures. There are few evaluations of these latest summarization models for long text like academic articles or patent document [14].

U.S. granted 382,559 patents in 2022, including 325,445 utility patents, 34,370 design patents, and 1138 plant patents [15]. Quickly understanding these patents is important to legal business such as intellectual property protection, product development, and fighting patent infringements. A U.S. patent document mainly contains the application information, followed by an abstract, prior arts, summary of the invention, which describes the invention and is the main part of the document. The length of the summary of the invention usually is around 10000 words. The description of the summary of the invention is also enhanced with figures and tables, which make the automated summarization of the document more challenging. However, the most important content of a patent document is the claims of the invention that are summarized in the summary of the invention. The length of the claims are normally around 1000 words, which can be handled by many automated text summarization models. Automated text summarization has been used for summarizing patent documents with limited success. [16] developed an extractive summarization system for patents using statistical and semantic features like term frequency and keyword matching. [17] fine-tuned BART for patent title generation from patent documents.

However, we found none of SOTA text summarization models can handle long text such as patent documents well in general. Some models such as GPT summarize the text mainly based on the beginning and end of the text but almost ignore the middle of the text [18]. Almost all models cannot take an input that is over a limit of the length of the text such as 1000 tokens (i.e. words and other symbols) [19]. Some models cannot appropriately handle of the non-uniform of the information in a document. For example, if an abstract and detailed description of the abstract are included in one document, then the detailed description could be totally ignored in the summarization [18]. None of the SOTA models can customize the level of details of the generated summary although the length of the summary can be customized, not the information. Therefore, it is necessary to conduct a systematic comparative evaluation of SOTA summarization models on long text. Our work aims to assess the quality of SOTA summarization models. We choose the patent document as the original text and investigate the performance of each model in selected text summarization quality evaluation criteria. Based on the evaluation results, we experiment and propose potential techniques for improving summarization quality in this domain. We conduct a comparative study of several SOTA summarization models including BART [10], PEGASUS [11], variations of T5, XLNet, BigBird, Longformer, and GPT-3.5, on U.S. patent documents. Our goal is to determine how well these general-purpose models summarize patents out-of-the-box and investigate techniques to improve their performance. Further, we propose and demonstrate a promising strategy to enhance the summarization of long texts by hierarchical decomposition.

In this work, we conduct a quality evaluation of SOTA text summarization models on around 1630 U.S. patent documents. The goal is to analyze the capabilities of current text summarization models on lengthy, structurally complex text and determine enable techniques for text summarization of patent documents. Specifically, we aim to answer following research questions:

- RQ1: What are the methods can be used to accurately evaluate the summarization models for long text?
- RQ2: Which summarization model has the highest quality for the summarization of long text?
- RQ3: How to improve the summarization of long text?

The main contributions of this research are summarized as follows:

- We summarize the state-of-the-art neural network models and evaluation metrics for text summarization.
- We conduct a comprehensive comparative study on the effectiveness of the summarization models for patent documents.
- We propose a prompt strategy for the quality improvement of patent summarization based on GPT-3.5.

## 2. Related Work

### 2.1 Text Summarization

Most current abstractive models rely on neural networks based sequence-to-sequence learning [4], [5]. In general, sequence-to-sequence models consist of an encoder and a decoder, where the encoder learns the contextualized representation of the input while the decoder attempts to reconstruct the encoded information in a left-to-right manner. Seq2seq summarization can be summarized into two main types of frameworks, RNN encoder-decoder [4] and Transformer encoder-decoder [5].

[20] is among the first RNN-based summarization models. They use a bidirectional RNN as the encoder to learn the input's presentation which is further enriched by POS tag and TFIDF feature embeddings. Meanwhile, the decoder is another unidirectional RNN that incorporates an attention mechanism to manage how the hidden states of the source text interact with the summary vocabulary. The RNN-based summarization models frequently encounter out-of-vocabulary (OOV) and word repetition issues. [21] proposes a pointer-generator network to handle the OOV problem through combining the base seq2seq model and a pointer network. The pointer network includes a soft switch $\in [0, 1]$ to decide whether it should generate the summary word by sampling from the vocabulary distribution or copy a word from the input sequence by sampling from the attention distribution. The coverage mechanism is additionally used to keep track of what has been summarized to prevent repetition [21], [20].

Abstractive summarization using Transformer encoder-decoder framework has rapidly advanced in recent years. Transformers with self-attention layers allow parallelization learning, solving the vanishing or explosion gradient of standard RNNs. It achieves SOTA performance in machine translation [5]. Given this success, this approach is promising in abstractive summarization. Currently, encoder-decoder Transformer models like BART [10] and PEGASUS [11] have achieved SOTA summarization results on short text. However, BART's and PEGARUS's maximum input length limit at 1024 tokens, making it unsuitable for summarizing long text. In addition, abstractive summarization often generates inaccurate or incoherent outputs. Some recent work has focused on improving faithfulness in abstractive summarization through techniques like factual knowledge graph [22], [6], and factual post-editing corrector [23], [24]. However, the problem is far from solved, especially for complex documents.

The major limitation of transformer models is the complexity of quadratic self-attention that grows rapidly with sequence length [13]. This has significantly impeded their effectiveness in summarizing long documents. The simplest approach is truncating the document from the head or tail to produce a short valid input. However, [14] proves that Transformers with this naive method is even worse than many unsupervised algorithms, such as TextRank, LSA, etc., for long text summarization. Models like Longformer [12] and BigBird [13] incorporate sparse attention mechanisms in the encoder to reduce the computational cost of standard self-attention operation [5]; therefore, it can handle longer contexts. Longformer can take up to 16k input tokens while Bird can support a sequence length of 4k tokens. Longformer Encoder-Decoder (LED) is a Transformer model that uses the BART architecture and Longformer's attention to handle longer context for sequence-to-sequence tasks. The proposed attention replaces the full self-attention in standard Transformers [5] with the attention pattern mechanism, including windowed, dilated, and global attention. The model achieved SOTA performance on arXiv dataset, a long text summarization dataset, surpassing BARD-base, Pegasus, and BigBird [12]. BigBird reduces the quadratic complexity by introducing a sparse attention mechanism combining random, windowed, and global attentions. BigBird leverages additional global tokens to apply full attention as Longformer. This mechanism allows BigBird to scale to longer sequences without significantly increasing computational resources. The model pretraining is continued from Pegasus [11] that is specified for abstractive summarization. The model performs better than base Transformers, Pegasus, BIGBIRD-RoBERTa, etc. on three long-text summarization datasets, including BigPatent, arXiv, and Pubmed [13]. Hierarchical summarization methods aim to capture the multifacet structure of long documents. [14] hypothesizes that lengthy documents like research papers may have a hierarchical structure comprising multiple facets such as purpose, method, findings, and values. Therefore, controlling the generated summary to cover information on those facets can result in better performance. They measure the alignment ROUGE score between facet-wise documents and references to select the most aligned sentences for inputting in BART. Their BART-Facet model performs better than the vanilla BART, indicating the value of hierarchical decomposition strategies for long text.

### 2.2 Quality Evaluation of Summaries

There are many studies on choosing the metrics for the quality evaluation of summaries. Some studies propose reference-based metrics, such as ROUGE [25], BLEU, BERTScore [26] and SummaC [27], QuestEval [28], as a viable option due to their ability to calculate a similarity percentage between a summary and its reference. QuestEval leverages question generation and answering to better evaluate factual consistency, while SummaC adapts natural language inference models to detect factual inconsistencies. Reference-based metrics are promising as they can achieve superior summary performance by using a question-answering approach [29]. Additional studies also showed that reference-based metrics such as BERTScore correlate more closely with human judgements [30].

In contrast, some studies claim that text overlap-based metrics, such as ROUGE and BLEU achieve the strongest correlation with human assessment. Graham [31] uses summary coverage computations and human coverage scores to assert that text overlap-based metrics are suitable for evaluation. However, studies also showed that BLEU, ROUGE, and BERTScore are not sufficient for the evaluation of summaries [32]. However, manual evaluation by human experts can be unstable [33]. The work introduced in [34] use readability metrics FRE, DCR and CLI to evaluate the performance of large-scale language models, specifically ChatGPT, in two controlled generation tasks. These tasks involve examining how well ChatGPT can adapt its output to different target audiences (experts vs. laymen) and writing styles (formal vs. informal). Additionally, it studied the evaluation of the faithfulness of the generated text and compared ChatGPT's performance with human-authored texts, which revealed that human-created variations in writing styles are significantly larger than those produced by ChatGPT. Article [35] found the challenge of ensuring factual consistency in summaries is due to the mismatch between the granularity of natural language inference datasets in sentence-level) and inconsistency detection tasks in the document level. To address this issue, the researchers propose a new method called SummaCConv and a new benhmark called Summac.
In summary, most summarization systems are automatically assessed by using the following popular metrics: ROUGE, BLEU, BERTScore, SummaC, and QuestEval. However, recent studies have identified issues with solely relying on these metrics, suggesting the need for human evaluation and improved metrics [36]. Our work analyzes both automatic metrics and human assessment to provide a comprehensive view of summarization quality.

## 3. Methodology

We evaluate summarization models including T5 base and its three variants, XLNet, BART, BigBird, PEGASUS, and GPT-3, using a corpus of 1630 patent documents. We used both semantic and linguistic criteria such as ROUGE, BLEU, and BERT Score to assess the generated summaries. Statistical analysis is used to compare model performance across metrics to determine their effectiveness in summarizing patents. Figure 1 shows the procedure of the experiment study.

### 3.1 Data Preparation

The dataset for this study consists of a corpus of 1630 patent documents collected through web scraping of Google Patents at https://patents.google.com. We developed a Python program to extract the full text of each patent and store it into a JSON file. The key information fields of patent number, abstract, and the summary of the invention including claims are extracted from each file and consolidated into a master Excel spreadsheet. The raw text extracted from patents contained

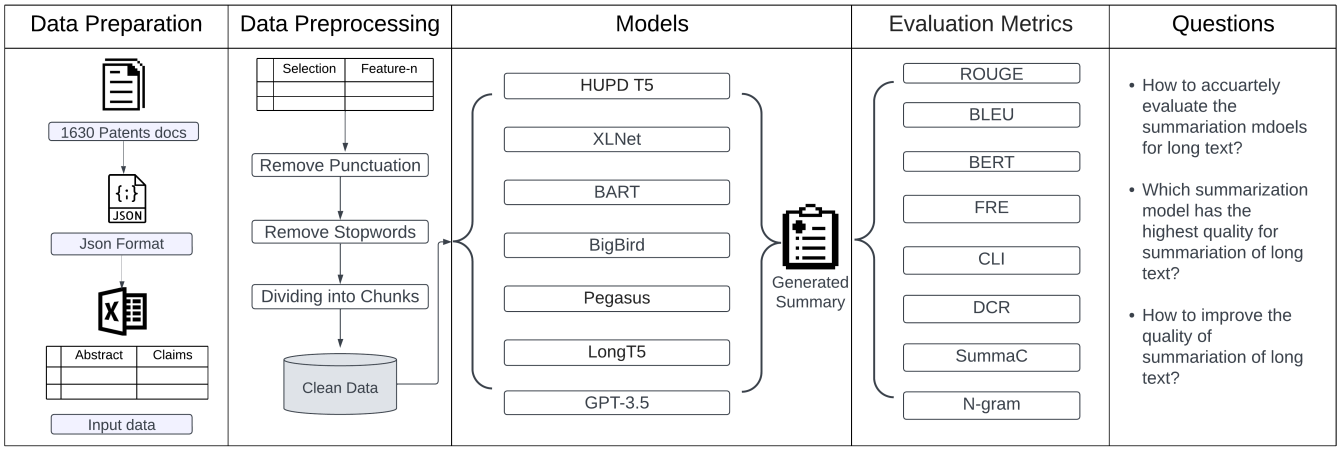

Figure 1. The comparison study of text summarization models

special characters such as #, , $, etc. which could potentially confuse the models. Therefore, these special characters are removed. Additionally, extra whitespaces and newline characters are removed. NLTK library's sentence tokenizer to used to conduct sentence segmentation for splitting text into constituent sentences. Any sentences consisting solely of patent section headers or numbers without useful surrounding context were removed from the corpus. Finally, all remaining text is converted to lowercase to standardize the case formatting. Overall, the preprocessing steps sanitize the raw text by removing confusing characters, inconsistent whitespace, irrelevant headers/numbers, and case inconsistencies. It produces a clean corpus of sentences tailored for summarization.

### 3.2 Summarization Models

In this section, we describe selected SOTA neural network models that have shown significant promise in text summarization. The models studied in this research include T5-base and its three variants, XLNet, BART, BigBird, Pegasus, and GPT-3.5. These models have been utilized across various applications including text summarization. These models are different in their architectures, training data, number of parameters, and application domains.

- **T5 Based Models** The T5 (Text-To-Text Transfer Transformer) models, developed by Google, are encoder-decoder transformers designed for a variety of natural language processing tasks [37]. They were used in various fields thanks to their capability to convert any language task into an essential text-to-text task.
  - **HUPD-T5** [38] Harvard University's Policy Department (HUPD) has tailored the T5 model for legal document summarization. There are two versions of this model, each with different sizes and computational efficiencies.
    - **HUPD-T5-Base:** HUPD-T5-Base is a variant of the T5 model, specifically fine-tuned on the HUPD legal dataset for tasks like legal document summarization. It has around 60 million parameters. Its training on a specialized legal dataset enables it to capture the nuances of legal language and deliver high-quality summaries of legal documents. This model is balanced between computational efficiency and performance. Despite it is smaller than the large variants, it still offers robust performance on legal summarization tasks. However, it requires more computational resources than smaller variants and does not capture as more fine-grained details as larger models.
    - **HUPD-T5-Small:** HUPD-T5-Small is a smaller variant of the T5 model fine-tuned specifically for legal document summarization with a focus on the most crucial legal points. It is trained on the HUPD legal dataset and has approximately 60 million parameters. Regarded for its computational efficiency, it possesses the capacity to succinctly summarize legal documents with efficacy. However, it may not be able to capture as many details or produce as nuanced summaries as larger models.
  - **Long-T5-TGlobal-Base-16384-Book-Summary-2:** Long-t5-tglobal-base-16384-book-summary-2 is an extended model in the T5 family [39]. It's trained with an expanded context window of 16384 tokens, enabling it to comprehend and summarize long text efficiently [40]. The number of parameters in this model is not specified but being a T5 base variant, it likely has around 220 million parameters. The model is trained on a large corpus of book summaries, providing it with a strong capability to digest and generate summaries of long text. However, its computational requirements are high especially during training.
- **XLNet:** an autoregressive transformer model, has 110 million parameters and trained on an assortment of datasets such as BooksCorpus, Wikipedia, and Giga5 [41]. Its distinguishing feature is the permutation-based training, allowing the model to capture bidirectional contexts and

understanding the intricate dependencies and relationships within the text, hence making it suitable for applications such as question answering and sentiment analysis. The capturing of the bidirectional context provide richer, more nuanced interpretations of text. However, it suffers from slower training and inference times.

- **Bidirectional and AutoRegressive Transformers (BART):** an encoder-decoder transformer model with 140 million parameters [10]. It has been pre-trained on a wide variety of data sources such as BookCorpus, Wikipedia, news articles, and stories. The model is popular for text summarization and question-answering tasks, thanks to its denoising autoencoder pretraining that helps generate more coherent and meaningful summaries. However, the requirement of large datasets for pretraining can be a drawback in scenarios where such data and computational resources.
- **BigBird:** a transformer model that pioneers the use of a sparse attention mechanism with 110 million parameters. It is trained on diverse data sources such as Wikipedia, Book-Corpus, and news articles. The introduction of the sparse attention mechanism enables BigBird to efficiently manage very long sequences, thus it reduces the computational complexity that is generally associated with processing long-range dependencies in text [13]. It is an excellent choice for tasks such as summarization and question answering that often require processing lengthy texts. Nonetheless, its sparse attention design might capture a diminished amount of context when juxtaposed with models utilizing complete attention mechanisms, potentially influencing the overall excellence of the model's outputs.
- **pegasus-x-large-booksum-1:** Pegasus [11] has a massive 568 million parameters. This encoder-decoder transformer model is notable for its pretraining objective, which is specifically optimized for summarization. It is trained on diverse datasets like C4, HugeNews, PubMed, and arXiv. It generates high-quality abstractive summaries and it is suitable for applications such as translation and question answering. However, the size of the model can pose challenges, as managing and deploying such a large model can be demanding in terms of computational resources and require considerable storage space.
- **GPT-3.5-turbo-16k:** [18] an evolution of the GPT model, is a computational heavyweight with 3.5 billion parameters. It is pretrained on a wide range of data sources, including WebText2, Common Crawl, Wikipedia, Reddit, and BooksCorpus. This model can be used for a variety of tasks, such as text completion, chat, and function calling. One of its notable features is its extended context window that can handle up to 16k tokens. This feature, along with its optimization for chat applications, allows it to maintain context over longer conversations and generate more accurate and coherent responses. However, the model is closed sourced and requires payment to generate each token.

## 3.3 Evaluation Metrics

Many metrics has been proposed for text summarization evaluation, however, not all of them are suitable for evaluating summarization on long text. We select and briefly introduce several recently used evaluation metrics which are appropriate for this study, including: ROUGE, Bleu Score, Bert Score, Flesch Reading Ease, Dale Chall Readability, Coleman-Liau Inde, N-gram, and SummaC.

### 3.3.1 ROUGE

The Rouge Score (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics designed to evaluate automatic summarization, especially for text. It's also frequently used in machine translation, text generation, and other natural language processing tasks where output can be compared to a reference text. ROUGE-N measures the number of N-gram matches between the system-generated summary and the reference summary. To calculate Rouge scores, one typically compares the generated summaries against one or more reference summaries. The calculation involves comparing n-grams (contiguous sequences of words) between the generated and reference summaries. Below are the metrics used in the ROUGE evaluation:

- **Overlap Score (ROUGE-1):** ROUGE-1 measures the unigram overlap between the candidate summary (generated summary) and the reference summary (ground truth summary). It calculates the proportion of overlapping unigrams (individual words) between the candidate and reference summaries. It considers only individual words and does not capture word order or context [42].
- **Coherence Score (ROUGE-2):** ROUGE-2 measures the bigram overlap between the candidate summary and the reference summary. It calculates the proportion of overlapping bigrams (consecutive pairs of words) between the candidate and reference summaries. It captures some level of word order and context by considering pairs of words together.
- **Informativeness Score (ROUGE-L):** ROUGE-L, also known as Longest Common Subsequence (LCS) based ROUGE, measures the longest common subsequence between the candidate summary and the reference summary. It finds the longest sequence of words that appear in the same order in both the candidate and reference summaries. ROUGE-L accounts for word order and captures the informativeness of the candidate summary by considering sequences of words rather than just individual words [42].

### 3.3.2 Bleu Score

The BLEU (Bilingual Evaluation Understudy) score is a widely-used metric for evaluating the quality of machine-generated text, especially in machine translation. The BLEU score quantifies the similarity between the machine-generated text and one or more reference texts. It considers the matching n-grams between the generated text and the reference text. The final BLEU score ranges from 0 to 1; 1 indicates a perfect match with the reference, while 0 indicates no overlap in n-grams.

### 3.3.3 BERT Score

BERT Score computes a similarity score between a generated text and a reference text by matching words in the two texts using contextual embeddings. BERTScore has been shown to correlate well with human judgment of text quality, and it has been used to evaluate a variety of text generation tasks, including machine translation, summarization, and question answering.

### 3.3.4 Flesch Reading Ease (FRE)

The Flesch Reading Ease (FRE) score assesses the readability of an English text by examining the sentence length and word length. It's calculated: $FRE = 206.835 - (1.015 * ASL) - (84.6 * ASW)$, where $ASL$ is the average sentence length and $ASW$ is the average number of syllables per word. The score typically ranges from 0 to 100. Higher scores indicate that the text is easier to read, while lower scores indicate that the text is more difficult to read.

### 3.3.5 Dale Chall Readability (DCR)

Dale-Chall Readability (DCR) score is another readability metric used to assess the readability of English text. It considers a set of familiar words and examines the sentence length to estimate the text's difficulty level. The DCR score is calculated: $DCR = 0.1579 * (PDW) + 0.0496 * (ASL)$, where $PDW$ is the percentage of difficult words in the text, and $ASL$ is the average sentence length. DCR provides an estimated percentage value representing the difficulty level. Lower DCR scores indicate higher difficulty.

### 3.3.6 Coleman-Liau Index (CLI)

Coleman-Liau Index (CLI) score is another readability metric used to assess the readability of English text. The CLI is calculated: $CLI = (0.0588 * L) - (0.296 * S) - 15.8$, where $L$ is the average number of characters per 100 words, and $S$ is the average number of sentences per 100 words. Higher scores suggest easier readability, while lower scores indicate more complexity.

### 3.3.7 N-gram

N-gram is used to assess the similarity or proximity between texts by examining sequences of words or characters called "n-grams."

### 3.3.8 SummaC

The SummaC measures the inconsistency between a summary and its source text. The score is calculated by comparing the summary to the source text and identifying any inconsistencies in the information according to Relevance Assessment, Decision-making Time, and Informativeness. The scores are then grouped into three ranges, with a score of 0-50% indicating low inconsistency, a score of 50-75% indicating medium inconsistency, and a score of 75-100% indicating high inconsistency.

## 4. Evaluation Results

We collected 1630 U.S. patents on communication and streaming technologies. Although a patent document includes long description of the invention details and many flow charts, but the most important content in a patent document for the summarization is its abstract and the claims of the invention. The abstract provides an overview of the invention, and claims are the invention details a summarization needs to capture. We hope the abstract provides a context for the summarization, and the claims provide information for the summarization. Therefore, we extract the abstract and the claims from each patent document. The comparison study of the text summarization models is conducted on different configurations of the abstract and the claims. For example, the input to a text summarization model could be: an abstract, all claims from a patent document, the combination of an abstract and its corresponding claims from a patent document. We also conduct summarization on summaries also. Therefore, the input also includes: the combination of the summary of an abstract and the summary of the corresponding claims from a patent document, the summary of the combination of the summary of an abstract and the summary of the corresponding claims from a patent document. Each summary is evaluated with the metrics we discussed before. Table I shows the evaluation result of each model on the output that consists of an abstract and its corresponding claims. We also evaluated the Roguge values of different configuration of the inputs such as only on claims or an abstract on models HUPD-T5-small and HUPD-T5-Base, but we found the best result is on the combined input. Therefore, we only evaluate the combined input in other models except GPT-3.5.

Based on Table I results, it is easily found that no single model stands out among the others. Model GPT-3.5 didn't perform the best on any metric. We also conducted an A/B testing on some metrics to see whether the difference in compared models is statistically significant. For the most comprehensive text summarization metric SummaC, A/B testing results show HUPD-t5-base is not statistically significantly better than HUPD-t5-small, or GPT-3.5-turbo16K ($p >= 0.05$). We also calculated the standard deviation of SummaC of each model on the total 1630 patent documents. The results show the SummaC is fairly uniform on all models in the dataset. Therefore, we conclude that the models we evaluated achieved impressive performance on text summarization of patent documents (such as the value of SummaC is from 0.79 to 0.94). However, none of these model achieved the performance as we expected. For example, when the abstract is included in the input document, the abstract is given much more attention than claims and the generated summary basically is a rewritten abstract and the claims are almost ignored. It is also impossible to customize the details of the generated summary. As soon as the input document includes an abstract or a generated summary with other content, then the generated summary will be close to the abstract or the summary. However, when an input document

TABLE I
EVALUATION SCORES OF ALL MODELS ON COMBINED INPUTS

| | HUPD_t5 Small | HUPD_t5 Base | XLNet | BART | BigBird | Pegasus | LongT5 | GPT-3.5 |
|---|---|---|---|---|---|---|---|---|
| BLEU | **0.38** | 0.22 | 0.002 | 0.07 | 0.0004 | 5.40 | 1.68 | 0.004 |
| BERT | **0.83** | 0.81 | 0.75 | 0.67 | 0.49 | 0.59 | 0.56 | 0.67 |
| FRE | 36.20 | **41.53** | 23.30 | 35.65 | 23.06 | 37.27 | 38.59 | 31.20 |
| CLI | 12.54 | 13.59 | 13.76 | 13.88 | **9.48** | 13.01 | 13.28 | 15.40 |
| DCR | 9.09 | 10.66 | 9.87 | 10.66 | **6.90** | 10.22 | 10.16 | 9.54 |
| SummaC | 0.84 | 0.93 | **0.94** | 0.86 | 0.79 | 0.87 | 0.86 | 0.91 |
| SummaC sd | 0.035 | 0.055 | 0.065 | 0.015 | 0.085 | 0.005 | 0.015 | 0.035 |
| N-Gram | 138.05 | 53.09 | **1873.96** | 53.09 | 441.63 | 274.13 | 269.47 | 942.98 |
| rouge-1 | **0.73** | 0.71 | 0.52 | 0.36 | 0.11 | 0.32 | 0.27 | 0.42 |
| rouge-2 | **0.65** | 0.61 | 0.39 | 0.22 | 0.03 | 0.07 | 0.04 | 0.21 |
| rouge-l | **0.73** | 0.71 | 0.52 | 0.36 | 0.10 | 0.29 | 0.24 | 0.40 |

*rouge-1: Overlap_Score['rouge-1']; rouge-2: Coherence_Score['rouge-2']; rouge-l: Informativeness_Score['rouge-l']. SummaC sd: standard deviation of SummaC. A combined input is the combination of the abstract and the claims in a patent document. The bold numbers represent the best value. The value is the average calculated from the 1630 patent document.

only includes claims, the summary generated from each model would be much more detail. Therefore, we experiment several ways for improving the summarization quality. Although GPT-3.5 didn't achieve the best performance, it performed better than most other models in all evaluation metrics and it is easy to prompting GPT-3.5 than other models. We also found GPT-3.5 can summarize the input consisting of abstract and claims better than other models when we evaluate it by people manually. We conduct the following experiments only on GPT-3.5 model.

### 4.1 Performance Improvement

We evaluated GPT-3.5 on individual abstracts, claims, combination of abstract and claims, combination of the summary of abstract and the summary of claims, and summary of the combination of the summary of abstract and the summary of claims (i.e. summary of the summary of combination of summaries) on each metric. Table II shows the results. According the results, it is easy to find that the performance on the summary of the summary of combination of summaries is the worst on almost every metric. Therefore, it doesn't make sense to simply repeat the summarization on a summary again and again since it may degrade the performance.

When the input includes only an abstract, then generated summary is a rewritten of the abstract. The invention details won't be summarized. When the input includes only claims, then generated summary doesn't include an overview of the patent. When the input includes both abstract and claims, then generated summary includes overview of the patent as well as summary of the claims although we hope the summary of claims could be in more details. Therefore, extracting appropriate content from long text such as patent document as the summarization input could be an easy way for improving the summarization quality and efficiency since GPT-3.5 could perform better in shorter version of a long document [18]. There are different ways for combing the abstract and the claims from a patent document. For example, we can keep the abstract in the beginning, followed by claims, or claims in the beginning, followed by abstract, or insert the abstract into the middle of claims. We found that the quality of the generated summaries is almost same based on SummaC (range from 0.94 to 0.95 on our ) or manual evaluation. Therefore, shuffling the paragraphs of an input won't improve the summarization.

Prompting GPT-3.5 is a straightforward way for improving the quality of summarization. However, the key is on how to prompt the model with effective prompts and procedure. We designed several prompts for the summarization of the combination of the abstract and claims:

1. Provide guiding instructions on what the summary should contain such as: *Focus on key technical elements and functions, capture core innovations and inventions, avoid repetition, maintain logical flow and structure.*

2. Provide an example high quality summary as demonstration such as: *Here is an example high quality summary: [example summary]. Based on the example of high quality, please write a high-quality combined summary of abstract and claims below. [abstract + claims].*

3. Iterate with slight variations of phrasing and compare outputs such as: *Summarize the key technical points of the following in a concise and structured manner: [abstract + claims]*, and then *Concisely summarize the core inventions and innovations described in the following: [abstract + claims].*

4. Abstraction level prompt such as: 1) Abstract summary:

TABLE II
EVALUATION SCORE OF GPT-3.5 ON DIFFERENT INPUTS

| | Abstract | Claims | Summary | $Summary^2$ | Combined |
|---|---|---|---|---|---|
| BLEU | **0.20** | 0.11 | 0.12 | 0.15 | 0.004 |
| BERT | **0.75** | 0.72 | 0.75 | 0.45 | 0.67 |
| FRE | 32.82 | 31.95 | 28.57 | **72.51** | 31.20 |
| CLI | 15.37 | 14.44 | 16.24 | **9.35** | 15.40 |
| DCR | 10.43 | **8.31** | 10.35 | 15.00 | 9.54 |
| SummaC | 0.94 | 0.94 | **0.95** | 0.55 | 0.91 |
| N-Gram | 70.64 | **2998.90** | 309.44 | 8.63 | 942.28 |
| rouge-1 | 0.58 | 0.56 | **0.58** | 0.17 | 0.42 |
| rouge-2 | 0.33 | 0.32 | **0.37** | 0.06 | 0.21 |
| rouge-l | 0.54 | 0.54 | **0.57** | 0.16 | 0.40 |

*rouge-1: Overlap_Score['rouge-1']; rouge-2: Coherence_Score['rouge-2']; rouge-l: Informativeness_Score['rouge-l']. Summary: Abstract summary + Claim summary. $Summary^2$: Summary of (Abstract summary + Claim summary). A combined input is the combination of the abstract and the claims in a patent document. The bold numbers represent the best value. The value is the average calculated from the 1630 patent document.

*Write an abstract summary of the following patent abstract and claims in 1-2 sentences focusing only on the core technical concept: [abstract + claims].* 2) High-level summary: *Write a high-level summary of the following patent abstract and claims in 2-3 sentences capturing the key technical functions and components: [abstract + claims].* 3) Detailed summary: *Write a detailed technical summary of the following patent abstract and claims in 4-5 sentences covering the essential steps, methods, components, and innovations: [abstract + claims].*
Experiment results show prompt using approaches 2 and 4 list above can significantly improve the summarization quality based on SummaC (the average is over than 9.50) and human manual evaluation.

### 4.2 Implications

From the results of patent document summarization experiments described earlier, we summarize the insights as follows:

- There is no "one size fits all" solution for text summarization on patent documents. Models should be selected based on different quality metrics of the summary.
- GPT-3.5 can summarize the input consisting of abstract and claims better than other models, indicating the potential of the model in long text summarization.
- Extracting appropriate content from patent documents as the summarization input could be a feasible approach for improving the summarization quality, as can be seen from our experiments using certain sections as input.
- Prompting strategy is the key to the success of text summarization for patent documents based on GPT-3.5. Few-shot learning and narrowing down the length of the output summary might be the most effective strategies for prompt engineering.

## 5. SUMMARY AND FUTURE WORK

In this work, we conducted a comparative evaluation of several SOTA summarization models on a corpus of patent documents. The goal is to analyze their capabilities in summarizing long, complex technical texts and identify techniques to improve quality. Overall, the T5 models designed specifically for legal texts (HUPD-T5) generated the most accurate summaries with highest content overlap with the reference summaries according to ROUGE metrics. The HUPD-T5 Small model performed the best, producing informative summaries tailored to capturing the key legal details. GPT-3.5 also showed the most promising results since it was well balanced on all metrics and achieved competitive ROUGE and highest BERT scores. Providing the models the full abstract and claims as input was found to produce better quality summaries than just the abstract or claims alone. The additional context enables capturing more technical details and nuances within the patent text. Prompting GPT-3.5 achieved the best quality of summarization with customized details. How to automate the prompting procedure for text summarization using GPT or similar large language models is a promising direction, which is part of our future work.

## ACKNOWLEDGMENT

We thank Karthik Puppala, master student from Department of Information Science at University of North Texas for collecting and pre-processing the data.

## REFERENCES

[1] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text summarization techniques: A brief survey," *International Journal of Advanced Computer Science and Applications (ijacsa)*, vol. 8, no. 10, 2017.
[2] D. Curcic, "Number of academic papers published per year," access on July 25, 2023. [Online]. Available: https://wordsrated.com/number-of-academic-papers-published-per-year/
[3] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert systems with applications*, vol. 165, p. 113679, 2021.

[4] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] Z. Cao, F. Wei, W. Li, and S. Li, "Faithful to the original: Fact aware neural abstractive summarization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[7] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1906–1919.

[8] J. He, W. Kryściński, B. McCann, N. Rajani, and C. Xiong, "Ctrlsum: Towards generic controllable text summarization," *arXiv preprint arXiv:2012.04281*, 2020.

[9] G. Bao and Y. Zhang, "Contextualized rewriting for text summarization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, 2021, pp. 12 544–12 553.

[10] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.

[11] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," 2020.

[12] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[13] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang *et al.*, "Big bird: Transformers for longer sequences," *Advances in neural information processing systems*, vol. 33, pp. 17 283–17 297, 2020.

[14] R. Meng, K. Thaker, L. Zhang, Y. Dong, X. Yuan, T. Wang, and D. He, "Bringing structure into summaries: a faceted summarization dataset for long scientific documents," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021, pp. 1080–1089.

[15] "2022 u.s. patent filings statistics, by mckee voorhees & sease plc," access on July 25, 2023. [Online]. Available: https://www.lexology.com/library/detail.aspx?g=1170d66d-63b8-4901-b819-e88c67916a2f

[16] V. Sharma, A. Behera, and R. McGregor, "Automated patent summarization using information extraction," in *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. IEEE, 2019, pp. 168–175.

[17] Z. Boukhers, O. Colas, and J.-G. Ganascia, "Abstractive text summarization techniques for patent title generation," in *Canadian Conference on Artificial Intelligence*. Springer, 2020, pp. 419–431.

[18] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," 2023.

[19] J. Ding, S. Ma, L. Dong, X. Zhang, S. Huang, W. Wang, N. Zheng, and F. Wei, "Longnet: Scaling transformers to 1,000,000,000 tokens," 2023.

[20] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 280–290.

[21] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1073–1083.

[22] Y. Lyu, C. Zhu, T. Xu, Z. Yin, and E. Chen, "Faithful abstractive summarization via fact-aware consistency-constrained transformer," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1410–1419.

[23] S. Chen, F. Zhang, K. Sone, and D. Roth, "Improving faithfulness in abstractive summarization with contrast candidate generation and selection," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 5935–5941.

[24] M. Cao, Y. Dong, J. Wu, and J. C. K. Cheung, "Factual error correction for abstractive summarization models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6251–6258.

[25] K. Ganesan, "Rouge 2.0: Updated and improved measures for evaluation of summarization tasks," *arXiv preprint arXiv:1803.01937*, 2018.

[26] V. Kieuvongngam, B. Tan, and Y. Niu, "Automatic text summarization of covid-19 medical research articles using bert and gpt-2," *arXiv preprint arXiv:2006.01997*, 2020.

[27] P. Laban, T. Schnabel, P. Bennett, and M. A. Hearst, "Summac: Re-visiting nli-based models for inconsistency detection in summarization," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 163–177, 2022.

[28] T. Scialom, P.-A. Dray, P. Gallinari, S. Lamprier, B. Piwowarski, J. Staiano, and A. Wang, "Questeval: Summarization asks for fact-based evaluation," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 6594–6604.

[29] D. Deutsch, T. Bedrax-Weiss, and D. Roth, “Towards question-answering as an automatic metric for evaluating the content quality of a summary,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 774–789, 2021.

[30] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “Bertscore: Evaluating text generation with bert,” *arXiv preprint arXiv:1904.09675*, 2019.

[31] Y. Graham, “Re-evaluating automatic summarization with bleu and 192 shades of rouge,” in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 128–137.

[32] N. Schluter, “The limits of automatic summarisation according to rouge,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2017, pp. 41–45.

[33] S. Mithun, L. Kosseim, and P. Perera, “Discrepancy between automatic and manual evaluation of summaries,” in *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, 2012, pp. 44–52.

[34] D. Pu and V. Demberg, “Chatgpt vs human-authored text: Insights into controllable text summarization and sentence style transfer,” 2023.

[35] P. Laban, T. Schnabel, P. N. Bennett, and M. A. Hearst, “Summac: Re-visiting nli-based models for inconsistency detection in summarization,” 2021.

[36] T. Sun, J. He, X. Qiu, and X.-J. Huang, “Bertscore is unfair: On social bias in language model-based metrics for text generation,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 3726–3739.

[37] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” 2020.

[38] M. Suzgun, L. Melas-Kyriazi, S. K. Sarkar, S. D. Kominers, and S. M. Shieber, “The harvard uspto patent dataset: A large-scale, well-structured, and multi-purpose corpus of patent applications,” *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, vol. 1, p. 3552–3565, 2022.

[39] H. Zhang, X. Liu, and J. Zhang, “Summit: Iterative text summarization via chatgpt,” 2023.

[40] Peter Szemraj, “long-t5-tglobal-base-16384-book-summary (revision 4b12bce),” 2022. [Online]. Available: https://huggingface.co/pszemraj/long-t5-tglobal-base-16384-book-summary

[41] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” 2020.

[42] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 2004, pp. 74–81.