

Evaluation of Question-Answering Based Text Summarization using LLM

Invited Paper

1st Junhua Ding[§]

Dept. of Information Science
University of North Texas
Denton, U.S.
junhua.ding@unt.edu

2nd Huyen Nguyen

Dept. of Information Science
University of North Texas
Denton, U.S.
huyennnguyen5@my.unt.edu

3rd Haihua Chen

Dept. of Information Science
University of North Texas
Denton, U.S.
haihua.chen@unt.edu

Abstract—Question-answering based text summarization can produce personalized and specific summaries; however, the primary challenge is the generation and selection of questions that users expect the summary to answer. Large language models (LLMs) provide an automatic method for generating these questions from the original text. By prompting the LLM to answer these selected questions based on the original text, high-quality summaries can be produced. In this paper, we experiment with an approach for question generation, selection, and text summarization using the LLM tool GPT4o. We also conduct a comparative study of existing summarization approaches and evaluation metrics to understand how to produce personalized and useful summaries. Based on the experiment results, we explain why question-answering based text summarization achieves better performance.

Index Terms—text summarization, generative AI, question-answering, large language model

I. INTRODUCTION

Text summarization aims to generate a concise summary of a text document, allowing users to quickly grasp the main content of lengthy texts. There are two major summarization methods: extractive and abstractive. These methods can also be combined as a hybrid approach. Extractive summarization selects sentences or phrases from the original text and combines them to create a summary. This method is faster, simpler, and often more accurate. However, it may result in summaries that are less fluent and coherent [1].

In contrast, abstractive summarization generates summaries by rephrasing the original text. This method can produce more fluent and coherent summaries, closer to those written by humans. However, it often suffers from “hallucination,” where the generated content is nonsensical or unfaithful to the source material [2], [3]. Abstractive summarization is also more challenging as it requires generating new sentences that convey the original text’s meaning [4].

The hybrid method combines both approaches, often producing an initial extractive summary and then refining it with abstractive techniques. Query-based text summarization [5] is

a type of hybrid method that generates summaries tailored to respond to specific queries. Question-answering text summarization is a special case of query-based summarization, where queries are replaced by questions, and the answers are used to generate the summary [6]. However, query and question generation and text summarization use the same technique in most cases and they may suffer the same problems. Reusing queries is not effective since the queries might not be enough to cover the original text, which would produce a summary that miss important content in the original text. LLMs offer a powerful tool to automatically produce queries and questions from a text document to cover the main content in the text. But do we still need to use query-based text summarization or can we just use the LLM tool to conduct the text summarization directly? In this paper, we conduct a comparative study to investigate the problem.

One can evaluate with or without a reference summary. Reference-based evaluation uses reference summaries to identify what content from the original document is important and then evaluates the generated summary based on the similarity between the generated one and the references. On the other hand, reference-free evaluation defines a model to capture important information in the original document and uses the model to evaluate the generated summary. However, high quality reference summaries are rarely available. Therefore, in this paper, we choose a reference-free evaluation method. In specific, we use a question-answering approach that was proposed in [6] for evaluating the quality of a summary since the ultimate goal of text summarization is to answer the questions that are answered in the original text. Using question-answering to evaluate text summarization involves generating a summary and formulating questions that cover key information from the original text. Answers are extracted from both the summary and the original text and compared using metrics like Exact Match and F1 Score. This process assesses the summary’s coverage, relevance, consistency, and coherence. By ensuring the summary accurately answers the questions, the evaluation provides a quantitative and qualitative measure of the summary’s quality.

[§]This research is partially supported by NSF grants #2231519, #2244259 and #2225229.

II. RELATED WORK

The breakthrough in deep learning have contributed to the progress and development of text summarization. In 2014, RNN Encoder-Decoder for Machine Translation was introduced and then it was enhanced with the attention and coverage mechanisms. Since then, the model was quickly adopted for abstractive summarization [7]. Transformer model [8] was introduced in 2017, a deep neural network with the self-attention mechanism, which achieved SOTA performances in most NLP tasks. Transformer is the backbone architecture of current pretrained Large language models (PLMs) including BERT, GPT and LLaMA. OpenAI released ChatGPT, a powerful chatbot based on GPT3, GPT3.5, and recently GPT4 and GPT4o [9]. These advancements in PLMs especially GPT have transformed text summarization also as they have demonstrated unprecedented capabilities in understanding context, semantics, and syntactic structures, enabling them to generate summaries that are concise, coherent, and human-like. Experiments on multiple news datasets, [10] even found that humans significantly prefer summaries generated by GPT to those written by humans or generated by small finetuned PLMs such as BART or T5. In this paper, the experiment is conducted on GPT4o, which is released on May 2024.

A. Text Summarization

Current neural networks based abstractive models is built for sequence-to-sequence learning [8]. A sequence-to-sequence model consists of an encoder and a decoder, where the encoder learns the contextualized representation of the input while the decoder reconstructs the encoded information.

Abstractive summarization using Transformer framework has rapidly advanced recently. It achieves SOTA performance in machine translation [8]. Currently, encoder-decoder Transformer models like BART [11] and PEGASUS [12] have achieved SOTA summarization results on short text. However, BART's and PEGASUS's maximum input length limit at 1024 tokens, making it unsuitable for summarizing long text. GPT4o can handle long text up to 128K tokens via API [13], which is sufficient for summarizing many documents. Even a long text cannot be summarized in one round, it is can be processed via multiple rounds. Therefore, the length limitation of GPT4o is not a problem for text summarization anymore. But abstractive summarization often generates inaccurate or incoherent outputs. It is also difficult to generated personalized summarization such as different details of specific content. Some recent work has focused on improving faithfulness in abstractive summarization through techniques like factual knowledge graph [14], cross checking [15], and factual post-editing corrector [16].

Query-based text summarization [5] focuses on producing summaries tailored to user queries of specific user needs. It is useful for applications requiring precise information retrieval, such as academic research, and legal services. Query-based text summarization, including question-answering based text summarization, represents a significant advancement in NLP, offering the ability to generate tailored summaries that address

specific user queries. Within this domain, question-answering based text summarization represents a specialized form where the query is typically in the form of a question. Extractive query-based summarization selects sentences or phrases from the source text that are most relevant to the query and then combine them. This relevance is often determined through metrics such as cosine similarity or neural ranking models [17]. Abstractive query-based summarization generates new sentences that succinctly address the query, and all models discussed in the abstractive summarization have been experimented for this purpose. Question-answering based summarization first identifies the relevant portion of the text that contains the answer and then generates a coherent summary that answers the question. This process can be implemented using extractive or abstractive methods [18]. However, the quality of the generated summaries heavily depends on the relevance and clarity of the queries or questions. Manual query generation involves human experts formulating queries or questions based on their understanding of the information need and the original text. Semi-automated methods combine human expertise with automated tools to generate queries. These methods can help scale query generation while maintaining a level of quality and relevance. For example, the template-based approach guides the generation of queries by providing predefined structures that users can fill in. Automated query generation uses algorithms and machine learning models to generate queries [19]. It is scalable and can handle large datasets but may face challenges in ensuring relevance and contextual accuracy. For example, information retrieval techniques and NLP models including sequence to sequence models, transformer models and reinforcement learning could be used for automated query generation. In this paper, we first used GPT4o to produce the questions and then manually refine and select the questions to be used for prompting GPT4o to generate the summary from the original text.

B. Evaluation of Text Summarization

The quality dimensions of text summarization include readability, fluency, consistency, faithfulness, relevance, and content quality. Readability refers to how easily the summary can be read, while fluency concerns the grammatical correctness. Consistency indicates how factually supported the summary is by the input document. Faithfulness assesses whether all information in the summary is present in the input. Relevance measures the selection of important content from the source document, and content quality refers to the inclusion of salient information. In practice, users may choose some of these dimensions to measure the text summarization quality.

There are many metrics for evaluating the text summarization quality dimensions, such as ROUGE [20], BLEU [21], BERTScore [22], [23], MoverScore [24], SummaC [25], and QuestEval [26]. They can be categorized into three groups: *text overlapping* metrics including ROUGE and BLEU, *vector-space distance* metrics including BERTScore and MoverScore, and *NLP task-based* metrics to measure the consistency between the generated summary and the reference including

SummaC and QuestEval. For example, QuestEval [26] leverages question generation and answering to evaluate factual consistency, while SummaC [25] adapts natural language inference (NLI) to detect factual inconsistencies.

Some studies claim that text-overlapping metrics achieve the strongest correlation with human assessment. [27] uses summary coverage computations and human coverage scores to assert that text overlap-based metrics are suitable for evaluation. However, other studies demonstrated that evaluation tools such as BLEU, ROUGE, and BERTScore are not suitable for the automatic evaluation of summaries [28]. Therefore, most studies combine different metrics, such as text-overlap-based (ROUGE, BLEU), vector-space-based (BERTscore, Mover-Score), and QA-based (QuestEval, SummaQA) to evaluate summarization performance.

Some reference-free metrics have recently been proposed including SummaQA [29], QuestEval [26], QAEval [30]. Most human evaluation methods are reference-free. But reference-based metrics have more advantages. For example, the question-answering approach as a reference-free method is able to detect hallucinations. Meanwhile, the reference-based approach for evaluation has limitations since it relies solely on the content of the reference summary, which may not always be of high quality.

In this research, we will use the question-answering approach to evaluate the text summarization. We first ensure the questions should cover the main content in the original document, then we need to select the questions that should be able only answered by the content in the original documents, but not by general knowledge. The evaluation value is the percentage of questions that are correctly answered by the generated summary.

III. EXPERIMENT STUDY

We conduct the text summarization using GPT4o for the patent documents that can be found in the website: patents.google.com. GPT4o can handle multi modal data including figures, tables, images and text in a document. In this experiment, we only evaluate the summarization against to the original text although GPT4o may refer to other data in the document. It would be intriguing to explore how GPT4o can handle non-text data and utilize the outcomes to improve text summarization of a document that contains substantial content in tables, figures, and images.

A. Experiment Setting

The experiment process is straightforward. We first upload a patent document in PDF format into GPT4o chat, then ask it to summarize the document based on prompts. We create a new chat in GPT4o and upload the same document each time when a new summary needs to be generated in case one may impact another. For the question-answering summarization, we first ask GPT4o to create questions that cover the main content of the document. We review these questions and then update the questions if needed. Selected questions are uploaded in the chat as prompts to find the answers. We ask GPT4o to find

the answers only based on the uploaded document. Finally, we create a summary based on the answers of the questions. We experimented different ways for producing the questions and prompting the GPT4o.

We searched patent documents from patents.google.com regarding “streaming services”. We chose U.S. patent document #US20210352125A1¹ as an example to illustrate the experiment since we are familiar with technology and work. The patent document includes an overview of the patent application such as the title, publication number, inventors, assignee, filed date, and abstract. Following the overview, it includes a group of drawings to illustrate the patent. After that, it is the text description of the patent including the following sections: title, field, background, summary, brief description of the drawings, detailed description. U.S. technical patents have the similar structure, but the description of each section could be different, and each may have different sections.

The patent #US20210352125A1 is about the devices, systems, and methods for converting or translating dynamic adaptive streaming over http (DASH) to http live streaming (HLS). The patent includes two aspects according to the summary described in the document: “a method ... receiving a media presentation description (mpd) file from a DASH server; extracting content from the mpd file; building a manifest file from the extracted content ; and providing the manifest file to an HLS client. ... a converter system ... a processor configured to load and execute instructions from a translator module; and said translator module configured to: receive a media presentation description (mpd) file from a DASH server; extract content from the mpd file; build a manifest file from the extracted content; and provide the manifest file to an HLS client.”

B. Simply Prompting GPT4o

We first just upload the patent document and then ask GPT4o to summarize the document:

Summarize the attached document. Try to read the figures and tables in addition to the text.

The output from GPT4o includes the following separated titled sections: the patent title, abstract, field of invention, background, summary, figures and tables, detailed description, and claims, which are copied from the original document except claims that described at the end of the original document without a separated section title (but may patent documents have a separated claim section. The output gives an overview of the patent without any technical details. One cannot find any specific invention or technical details. It doesn’t cover any more information that is covered in the summary section of the original document.

We ask GPT4o to write a summary based on above output:

Based on your output above, write a summary without the separated sections, but focus on readability and cohesion among sentences. It should just like a long abstract.

¹<https://patents.google.com/patent/US20210352125A1/en?qoq=US20210352125A>

Although the summary does not add new information, it is better structured and its readability is significantly improved. For example, the key of the invention is the implementation of the adaptive streaming through leveraging Adaptive Bit Rate (ABR) streaming technology. In the first output, it says:

Adaptive Bit Rate (ABR) streaming is a technology that breaks content into small file segments, each containing a short playback interval. This allows the stream to adapt to available data rates..

The corresponding information in the summary is:

The invention leverages Adaptive Bit Rate (ABR) streaming technology, which segments content into small files, allowing streams to adapt to available data rates, thereby ensuring continuous playback without rebuffering.

The later one is more related to the invention, but the earlier is only a definition.

We now ask GPT4o to describe the technical details explicitly:

Summarize the attached patent document, and give the technical details of the invention.

The output starts with “summary of the invention”, followed by the “Technical Details of the Invention”, which includes subsections: field of invention, background, invention details, example embodiments, and claims. It gives a description of the working process, and the definition of most of the functions. But it doesn’t offer useful technical details that distinguish this invention. The summary generated based on the output is longer than the first version, but it only adds more definition of the functions without real technical details. We expect the output would give details on the implementation of ABR and how it is used for converting one protocol to another. It appears that GPT-4 is unable to distinguish between the primary invention and the supporting techniques.

Of course, the details of the output and summary generated from GPT4o could be slightly different each time even with the same prompt, but the main content is always the same.

If we ask GPT4o to describe the specific technical details, what might happen? To explore this, we experimented by prompting GPT4o with key phrases.

C. Prompting GPT4o with Key Phrases

We first ask GPT4o to extract the most important technical phrases:

Extract the 10 most important key technical terms from the attached patent document.

It gives the ten terms with definition of each such as DASH, HLS, MPD, Manifest File, ARB, etc. If we ask it to extract the top 20 phrases, then the first 6 is same to the those in top 10, others could be same or different. If run the same prompt again, the result could be slightly different, but most important ones are always the same. We believe the extracted phrases could be good enough for forming prompt to generate summary with specific technical details that is missing in the above experiments.

We ask GPT4o to summarize the patent document with technical details described by the top technical terms:

Summarize the attached patent document. Give the technical details on how converting or translating Dynamic Adaptive Streaming over DASH to HLS Streaming and vice versa using ABR steaming technique.

The output is organized with the technical details, which first give two sections: one is conversion process including the system components, which explains the process from receiving the file from DASH server to converting them for adaptive streaming, and system architecture. It then gives an example process to explain the dynamic bitrate adjustment. It well explains how the invention works and provide the details on the implementation of the conversion process, which is the most important invention in the patent. However, the previous version covers more content in the original document.

In the new generated summary, the first graph adds “*This is achieved using Adaptive Bitrate (ABR) streaming techniques to ensure seamless media delivery under varying network conditions.*” to emphasize the implementation. Then it adds the conversion process: “*The conversion process begins with the system receiving a ... MPD file from a DASH server. The MPD file, ..., is then parsed Using the extracted content, the system builds a manifest file compatible with HLS, The system supports ABR streaming, allowing for dynamic adjustment of media quality Initially, the system streams ..., continuously ... to dynamically adjust the bitrate, ensuring optimal playback without buffering.*”. It also describes the example process in the summary. One of the author of this paper reviewed and manually summarized more than 5000 streaming technical patents before and believed this version of summary is better than manually created patent summarization in readability and technical details.

However, if we only give the top key terms such as:

Summarize the attached patent document. Give the technical details on DASH, HLS, and ABR.

in the prompt instead of a clear task such as:

how converting or translating Dynamic Adaptive Streaming over DASH to HLS Streaming and vice versa using ABR steaming technique.

the output and the generated summary are not much different to the ones that are generated without these terms except they give the definitions of DASH, HLS and ABR. Therefore, it is important to know which questions should ask for the GPT4o.

D. Prompting GPT4o with Questions

We first ask GPT4o to create questions:

Generate questions that cover the significant content in the attached document, and these questions should be able to be answered by the document.”

GPT4o generated 20 questions, and all of these questions are related to the patent document and covered the main content. It is impossible to produce questions to cover all details in a document since the number of possible details could be infinite. GPT4o could produce slightly different questions with the same prompt each time, but the difference won’t impact anything.

Here is a list of the 20 questions that GPT4o generated:

- 1) What is the primary purpose of the systems and methods described in the patent document?
- 2) Describe the process of translating dynamic adaptive streaming over HTTP (DASH) to HTTP live streaming (HLS) as outlined in the patent.
- 3) What components make up the converter system mentioned in the patent, and what are their functions?
- 4) How does the converter system handle media presentation description (mpd) files from a DASH server?
- 5) What are the steps involved in the converter system providing a manifest file to an HLS client?
- 6) How does the patent describe the handling of adaptive bit rate (ABR) streaming for an HLS client?
- 7) Explain the role of the adaptive algorithm module within the converter system.
- 8) What does the trick play module do in the converter system?
- 9) How does the patent propose handling trick play modes like pause, fast forward, and rewind?
- 10) How does the converter system adjust the content stream based on network conditions?
- 11) What is the significance of the DASH metrics module in the converter system?
- 12) How does the converter system ensure compatibility between different media container formats and DRM schemes?
- 13) What are the benefits of using adaptive bit rate (ABR) streaming as described in the patent?
- 14) What is the functional block diagram of the converter system as illustrated in FIG. 10?
- 15) Explain the process depicted in FIG. 9 for translating from DASH to HLS.
- 16) Describe the optional step for content and DRM conversion in the process of translating from DASH to HLS.
- 17) What are the steps involved in the converter system's interaction with the HLS client, from requesting a content list to displaying the audiovisual media asset?
- 18) How does the converter system manage bandwidth estimation and selection of the appropriate media segments?
- 19) What security measures are mentioned in the patent for the communication between client components and server components?
- 20) Describe the role of the web server module within the converter system.

There are some almost same questions such as Questions 8 and 9, some of these questions are not necessary such as Questions 14 and 15, and some of these questions should be combined to ensure the entire process could be answered by one question. For example, Questions 4 and 5 on the conversion process are an example. If we are not sure which questions should be kept, then we may just start with all questions and then update these questions based on the answers of the these question given by GPT4o.

We ask GPT4o to answer these questions based on only the

patent document:

Generate the answers of all of these questions based on the attached patent document.

GPT4o gives the answer for each question. Each question is correctly answered without technical details. Since these answers only give fragmented information, it is difficult to understand the working principle and entire working process of the invention. For example, the answers for questions 8 and 9 are:

Answer of question 8: *The trick play module handles user requests for fast forward, rewind, and other playback controls by downloading the appropriate segments from the DASH server and making them available to the HLS client.*

Answer of question 9: *Trick play modes are handled by the trick play module, which translates requests from the HLS client into corresponding DASH requests and provides the necessary segments for playback.*

It is necessary to combine the two answers together to understand how the trick play works. For some users, more questions could be asked further such as:

Question (combine questions 8 and 9): *How does the trick play work from server side to the client side described in the patent?*

The answer includes the working process and implementation details. It explains the process starting with "Client Request", then "Converter System Role", "Request Translation", "Communication with DASH Server", ..., "Continuous Adjustment", and ending with "Completion". The answer would be useful to understand the technique.

We may rephrase the question 1 to find the most important invention of the patent: *What is the most important invention in the attached patent document?*

The original answer of question 1 is: *The primary purpose is to convert or translate Dynamic Adaptive Streaming over HTTP (DASH) to HTTP Live Streaming (HLS) and vice versa.*

The answer of the updated question is much more detailed: *is a system and method for converting or translating Dynamic Adaptive Streaming over HTTP (DASH) to HTTP Live Streaming (HLS) and vice versa. This invention addresses the compatibility issues between these two popular streaming protocols, enabling seamless streaming across different platforms that support either DASH or HLS.* Then it explains the key components and functions of the invention: Translator Module, Adaptive Algorithm Module, Trick Play Module, DASH Metrics Module, and Content and DRM Conversion.

Since the MPD file is important to above process, one may further ask the details about MPD:

How the HLS client to use the MPD file that is built on DASH MPD file?

GPT4o gives details on how the MPD file is used during the conversion process, but it doesn't explicitly mention the ABR, which is the key enable technique of the invention. One may ask:

How the ABR is used during the above process?

The answer explains how ABR is implemented for converting DASH to HLS based on MPD.

How can we produce the summary based on the answers of these questions? We prompt GPT4o with the input:

Produce a long abstract of the attached patent document through combining all answers above.

GPT4o gives a long summary that cover the purposes of the invention, the working principle, enabling technique, and the benefits. It doesn't simply combine the answers, but generate the summary using the questions and answers as guideline to structure the summary. The summary is almost same to the one produced through prompting with the key phrases:

Summarize the attached patent document, except it doesn't include an example process. Give the technical details on how converting or translating Dynamic Adaptive Streaming over DASH to HLS Streaming and vice versa using ABR steaming technique.

We can also enter all answers into GPT4o chat and ask it produce a summary via combining the answers:

Write a long summary based on the following input. Ensure the readability and cohesion among paragraphs.

The output is longer but without any additional technical details except it gives more definitions of some terms that were not defined in the previous one.

Based on above experiment results, we found question-answering based text summarization can well summarize the original document. It is easy to produce customized summary covering different content in different details. However, text summarization with key phrases also can produce the summarization with similar quality as soon as the prompting input is well formed. We believe the question-answering based text summarization provide a guideline for users to access needed information and for GPT4o to produce user needed summary.

In the following section, we discuss how to evaluate the quality of the generated summary and how to use the evaluation result to improve the text summarization quality.

IV. QUALITY EVALUATION OF TEXT SUMMARIZATION

Researchers have proposed many quality evaluation metrics for evaluating text summarization. Most of these metrics are adopted from the quality evaluation metrics for NLP tasks. Quality evaluation results can be used for guiding the acceptance or improvement of the text summarization. Depending on the evaluation purposes such as accuracy, coverage, consistence, or readability, different metrics could be used. In this paper, we utilize a question-based evaluation approach, measuring the effectiveness by the percentage of questions correctly answered by the generated summary. Our goal is to assess how well the summarization addresses users' questions. We also evaluate some of the widely used metrics for a reference purpose.

A. Evaluation Metrics

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [31] is a set of metrics designed to evaluate text summarization through comparing it to a reference. ROUGE-N measures the number of N-gram matches between the system-generated summary and the reference summary. Its score

includes precision and recall value. For example, ROUGE-1 refers to the overlap of uni-gram (each word) between the system and reference summaries, while ROUGE-2 refers to the overlap of bi-grams (two consecutive words).

BLEU (Bilingual Evaluation Understudy) score quantifies the similarity between the machine-generated text and one or more references. It measures the matching n-grams between the generated text and the reference. BLEU combines the scores for different n-grams (usually 1 to 4) into a single score by calculating the geometric mean of the modified precision counts [21].

BERTScore computes a similarity score for each token pair between the generated text and the reference using BERT embeddings. BERTScore has been shown to correlate well with human judgment of text quality [22]. Its scores such as Precision, Recall, and F1, range from 0 to 1, with higher scores indicating better performance.

SummaC measures the consistency between a summary and original text. The score is calculated by comparing the generated summary to the source text by identifying any inconsistencies between them. SummaC effectively utilizes NLI models for inconsistency detection by segmenting documents into sentence units and aggregating scores between pairs of sentences [25].

FRE (Flesch Reading Ease) score [32] assesses the readability of an English text by examining the sentence length and word length. However, the metric mainly focuses on sentence and word length and does not consider other factors such as content, coherence, or text structure which also impact readability.

DCR (Dale Chall Readability) 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. It primarily considers sentence length and the presence of difficult words but does not account for factors like content, coherence, or text structure, which also influence readability.

B. Evaluation Results

We first manually create a summary as a reference for each patent document to be summarized. Then we calculate the ROUGE, BLEU and BERTScore for each generated summary. Every generated summary for all 10 patent documents we tested with different prompting strategies by GPT4o achieved almost perfect scores for all three metrics. In other words, these metrics cannot distinguish the quality among all of these summaries. It is the same for SummaC. We didn't calculate the FRE or DCR score, but we read all of these summaries and each summary is well structured and written without any obvious errors. We believe question-based evaluation could be better evaluate the text summarization.

We first generate questions from the original document and then select the questions to check how the corresponding summary can correctly answer these questions.

We ask GPT4o to produce the questions:

Generate questions from the attached patent document. These questions are used to check the understanding of the invention, its applications, and its technical details..

GPT4o produced 20 questions, and most of these questions are similar to those we used to produce the summary. These questions will cover the main content of the patent document. Here is the list of the questions.

- 1) What is the main purpose of the invention described in the patent?
- 2) How does the system facilitate interoperability between DASH and HLS streaming protocols?
- 3) What role does the Adaptive Bitrate (ABR) streaming technique play in the invention?
- 4) Describe the process of receiving and extracting content in the conversion system.
- 5) How are manifest files built from the extracted content? What information do they include?
- 6) What are the components of the converter system, and what functions do they perform?
- 7) Explain the function of the translator module in the conversion process.
- 8) What is the significance of the adaptive algorithm module within the system?
- 9) How does the trick play module enhance the functionality of the streaming system?
- 10) What performance metrics does the DASH metrics module provide to the DASH server?
- 11) Why is the memory module important in the converter system?
- 12) How does the web server module contribute to the overall functionality of the system?
- 13) Describe the system architecture, including the roles of the DASH server, the converter, and the HLS client.
- 14) In what ways can the converter be integrated within the client device or act as a standalone server?
- 15) Detail the step-by-step process of converting DASH to HLS streaming from the perspective of the HLS client.
- 16) How does the converter handle changes in network conditions to ensure continuous playback?
- 17) What happens during the dynamic adjustment of the stream's bitrate?
- 18) What challenges does this invention address in the field of digital network data transmission?
- 19) What are the benefits of using this system for streaming media delivery?
- 20) How does the system handle URL requests for media segments, and what role does caching play in this process?

We manually check each summary to find which question can be answered by the summary, and then we ask GPT4o to check it again. We enter the summary in the chat with the prompt:

Can the following summary and only the summary sufficiently answer these questions?.

GPT4o gives the answer for each question. If the summary cannot explicitly answer a question, it is considered as not

having answered it, even if the answer may be implied within the summary. Consequently, this question is counted as unanswered. For example, the summary generated used the question-answering based text summarization discussed in this paper can correctly answer all these questions (i.e. 100%), and the summary generated using the simply prompting approach cannot answer questions 8, 10, 11 and 12. But answers to questions 8, 10 and 11 are important to understand the patent. The percentage of correctly answer these questions is 80%. We conducted the experiment on 9 other patent documents and found the similar result. However, the results for the summaries that are generated with prompting with key phrases are mixing, it is depending on the quality of the prompt forming with the key phrases. Therefore, we believe that question-answering based text summarization using GPT-4 could represent the future direction for text summarization.

C. Discussion

It is evident that GPT-4 offers a highly effective tool for automated text summarization. Its capabilities extend beyond just text summarization, as it can also efficiently summarize documents that include a mix of text, figures, tables, and images. The patent documents discussed in this paper contain figures, and GPT-4 can extract accurate information from these figures, though we did not evaluate its performance in this regard. This is one reason why the metrics discussed in IV-A are unsuitable for evaluating summarization quality, as they are defined solely for text. However, the question-based evaluation approach is more applicable. There are several reasons we chose patent documents for our case study. First, we are familiar with this type of work, having manually created over 5,000 summaries of patent documents. Second, patent application documents are well-structured, with the main points, such as the soundness and uniqueness of the invention, clearly presented. This makes it relatively straightforward to evaluate the quality of the summarization. We also experimented with summarization strategies for other legal documents, and found that question-answering based text summarization using GPT3, GPT4, and GPT4o consistently achieved the best performance.

Why question-answering based text summarization performed better? Question-answering may offer a superior method for extracting information from documents, providing targeted and relevant summaries tailored to specific user needs. By focusing on specific questions, the summarization ensures that the extracted information directly addresses the core aspects of the document, filtering out irrelevant details and emphasizing the most pertinent content.

The flexibility and customization afforded by question-answering based summarization allow it to adapt to the varying needs of different users. Each user may have unique questions or areas of interest, and this method can dynamically adjust to provide summaries that are highly relevant and useful. It results in clearer and more focused content, reducing the likelihood of extraneous information that might clutter traditional summaries.

V. SUMMARY AND FUTURE WORK

This paper explores question-answering based text summarization using LLM GPT4o to create personalized summaries. We compare different summarization strategies on patent documents including simply prompting, prompting with key phrases, and prompting with question-answering with GPT4o. The question-answering based text summarization leverages GPT4o to generate questions from the original text and using these questions to produce summaries. Experiments show that GPT4o can create high-quality summaries by addressing user-specific questions, improving relevance and coverage.

The evaluation uses a question-answering approach, measuring the percentage of questions correctly answered by the summaries, proving more effective than traditional metrics like ROUGE, BLEU, and BERTScore. Future work aims to extend GPT4o's capabilities to handle non-textual data, apply question-answering based summarization across various domains, and improve automatic question generation and selection algorithms. Additionally, refining evaluation metrics considering non-text summarization will further enhance summarization using LLMs, making it more robust for diverse applications.

REFERENCES

- [1] A. Tandel, B. Modi, P. Gupta, S. Wagle, and S. Khedkar, "Multi-document text summarization - a survey," in *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, 2016, pp. 331–334.
- [2] M. Cao, "A survey on neural abstractive summarization methods and factual consistency of summarization," *arXiv preprint arXiv:2204.09519*, 2022.
- [3] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.
- [4] 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.
- [5] H. Yu, "Survey of query-based text summarization," 2022.
- [6] M. Eyal, T. Baumel, and M. Elhadad, "Question answering as an automatic evaluation metric for news article summarization," *arXiv preprint arXiv:1906.00318*, 2019.
- [7] W. Zeng, W. Luo, S. Fidler, and R. Urtasun, "Efficient summarization with read-again and copy mechanism," *arXiv preprint arXiv:1611.03382*, 2016.
- [8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [9] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.
- [10] X. Pu, M. Gao, and X. Wan, "Summarization is (almost) dead," *arXiv preprint arXiv:2309.09558*, 2023.
- [11] 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.
- [12] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 328–11 339.
- [13] OpenAI, "Introducing gpt-4o: our fastest and most affordable flagship model," 2024. [Online]. Available: <https://platform.openai.com/docs/models/gpt-4o>
- [14] 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.
- [15] S. Tonmoy, S. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das, "A comprehensive survey of hallucination mitigation techniques in large language models," *arXiv preprint arXiv:2401.01313*, 2024.
- [16] 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.
- [17] A. K. Yadav, Ranvijay, R. S. Yadav, and A. K. Maurya, "State-of-the-art approach to extractive text summarization: a comprehensive review," *Multimedia Tools and Applications*, vol. 82, no. 19, pp. 29 135–29 197, 2023.
- [18] D. O. Cajueiro, A. G. Nery, I. Tavares, M. K. De Melo, S. A. d. Reis, L. Weigang, and V. R. Celestino, "A comprehensive review of automatic text summarization techniques: method, data, evaluation and coding," *arXiv preprint arXiv:2301.03403*, 2023.
- [19] J. Vig, A. R. Fabbri, W. Kryściński, C.-S. Wu, and W. Liu, "Exploring neural models for query-focused summarization," *arXiv preprint arXiv:2112.07637*, 2021.
- [20] L. Chin-Yew, "Rouge: A package for automatic evaluation of summaries," in *Proceedings of the Workshop on Text Summarization Branches Out*, 2004, 2004.
- [21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
- [22] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.
- [23] 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.
- [24] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, "Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.
- [25] P. Laban, T. Schnabel, P. Bennett, and M. A. Hearst, "Summac: Revisiting nli-based models for inconsistency detection in summarization," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 163–177, 2022.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] T. Scialom, S. Lamprier, B. Piwowarski, and J. Staiano, "Answers unite! unsupervised metrics for reinforced summarization models," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3246–3256.
- [30] 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.
- [31] 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.
- [32] R. Flesch, "How to write plain english: Let's start with the formula," *University of Canterbury*, 1979.