PresentAI: Automated Generation of Slides and Accompanying Speech

Anonymous ACL submission

Abstract

In today's fast-paced world, presentations have become an essential tool for communication in both online and offline meetings. The crafting of a compelling presentation requires significant time and effort, from gathering key insights to designing slides that convey information clearly and concisely. However, despite the wealth of resources available, people often find themselves manually extracting crucial points, analyzing data, and organizing content in a way that ensures clarity and impact. Furthermore, a successful presentation goes beyond just the slides; it demands rehearsal and the ability to weave a captivating narrative to fully engage the audience. We introduce PresentAI, a pipeline used to generate slides from general documents, going beyond just research papers, which also automates the oral delivery of the generated slides. PresentAI analyzes user documents to create a dynamic, engaging presentation with an AI-generated voice. Additionally, we developed an LLM-based evaluation metric to assess our pipeline across three critical dimensions of presentations: relevance, coherence, and redundancy.

1 Introduction

Presentations have become indispensable in academic, professional, and business contexts for effectively communicating complex ideas. They help visualize information, making it easier for audiences to absorb key takeaways. However, preparing these presentations, along with oral delivery, can be a challenging and time-consuming task, requiring numerous rehearsals and careful attention to timing. Creating presentation slides from document is a multi-step process that include: 1) defining the purpose of your presentation and outlining the main points to ensure clarity and focus. 2) selecting a clean, professional template and maintaining consistency in fonts, colors, and layout across all slides. 3) adding the main ideas using bullet points, visuals, and relevant media like images and charts to support the content (Sarter, 2006) and 4) organizing the content to focus on one idea per slide while ensuring a logical flow of information throughout the presentation (Green, 2021). Automation of this process saves time, ensures consistent delivery, and reduces the burden on presenters.

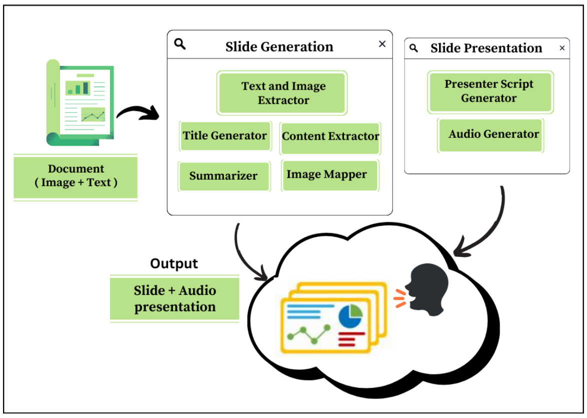


Figure 1: Overview of the PresentAI pipeline. It takes a user-provided document as input and generates presentation slides along with AI-generated voice narration.

Numerous studies have focused on automating the slide generation process. For instance, recent research by (Mondal et al., 2024) explores the use of LLMs for generating slides. One challenge identified in such work is the issue of content overlap due to fixed slide generation limits. Our approach builds on these insights by designing a flexible pipeline that generates upto 8-10 slides, ensuring the model only covers the necessary topics and avoids repetition where content is limited. Other works (Bandyopadhyay et al., 2024) could also struggle with content overlap between slides when sections are too similar.

Prior approaches such as (Winters and Mathewson, 2019) utilize rule-based heuristics to extract content for slides, while others, like D2S (Sun et al., 2021), treat document-to-slide generation as a query-based summarization task. Several studies, including (Sefid et al., 2019) and (Hu and Wan, 2015), focus on academic papers with well-defined structures, while (Fu et al., 2022) proposes a trainable sequence-to-sequence model, which requires large amounts of labeled document-to-slide data, making it difficult to scale. Despite valuable contributions, these approaches have several limitations: 1) The need for manual captioning of images to extract meaningful content from visuals. 2) Number of slides generated in above works is fixed, leading to redundancy when content is sparse. 3) Non-textual content such as images and graphs requires manual mapping to the relevant slides.

Despite advances in automated slide generation, the delivery of these slides remains a significant challenge. A successful presentation involves not just the content, but also the ability to deliver it effectively, maintaining audience engagement, and ensuring smooth timing (Ho et al., 2023). But what if the entire process—both content creation and delivery—could be fully automated? This is where our innovative pipeline, PresentAI, comes into play.

PresentAI addresses these challenges by introducing two core modules: slide generation and slide presentation. As shown in Figure 1, the slide generation module automatically generates slide titles and corresponding content based on the document, ensuring a structured and coherent presentation. The pipeline is versatile, working with both LLMs and multimodals, with multimodals processing both text and images for slide generation. For LLM-based approach, users must provide captioned images in the document. The second key component, the slide presentation module, generates a script for each slide based on the content and converts it into speech using AI voice synthesis.

Contributions: To the best of our knowledge, no prior work fully automates both the content generation and the delivery of a presentation using AI-generated voice. Our approach is the first to offer both slide generation and AI-powered audio delivery modules. While recent innovations such as NotebookLlama (Meta, 2024) and NotebookLM (Google, 2024) focus on document delivery in a podcast-style format, they do not address the complete automation of presentation delivery. In summary, our key contributions include: 1) a modular, AI-based pipeline for the automated generation and delivery of presentation slides. 2) a novel image mapping module that automates the process of mapping relevant images to corresponding slides. 3) a slide presentation module that generates a script for each slide and converts it into AI-driven audio. 4) an evaluation framework using LLM-based methods to assess key aspects of presentation quality—coherence, relevance, and redundancy.

We plan to release our code upon publication. These are made available for reviewing purposes in the supplementary material.

2 Architecture

The architecture of PresentAI comprises of two modules: Slide Generation and Slide Presentation as shown in Figure 1 and Figure 2 (Appendix). These modules work in coordination to automate both the content creation and the delivery of presentation slides. Throughout the development of each module, we continuously monitored output quality and iteratively refined the prompts to optimize performance.

2.1 Slide Generation

This module is responsible for transforming the input document into structured slides. It uses either LLM or multimodal(capable of processing both text and images) to generate the content. This module consists of five key sub-modules, each serving a specific function:

Image and Text Extractor: This sub-module is tasked with separating textual content from images in the input document. It ensures that the relevant text and images are properly processed for the subsequent stages, enabling flexibility in content generation depending on whether the model used is text-based or multimodal.

Title Generator: This sub-module creates up to 8-10 slide titles (T_i) based on the document's content. It considers the intended audience (e.g., technical or non-technical) and uses an LLM or multimodal to generate concise and relevant slide titles. This helps to tailor the presentation's focus and ensures that the generated slides align with the audience's level of understanding (Mondal et al., 2024).

Content Extractor: In this sub-module, the model analyzes the extracted text and the generated titles (T_i) to identify and extract the most relevant content (C_i) for each slide . The extraction process is guided by specific prompts (Biswas and Talukdar, 2024) which are given in the Appendix

Method	Coherence	Redundancy	Relevance	Average
D2S	7.42 ± 0.05	6.11 ± 0.14	8.48 ± 0.05	7.34 ± 0.08
GPT-Flat	8.58 ± 0.04	8.22 ± 0.09	9.46 ± 0.04	8.75 ± 0.06
GPT-COT	8.61 ± 0.04	8.03 ± 0.07	9.50 ± 0.04	8.71 ± 0.05
GPT-Cons	8.64 ± 0.04	7.98 ± 0.06	9.65 ± 0.05	8.76 ± 0.05
Qwen-PresentAI (ours)	8.65 ± 0.05	**8.35 ± 0.08**	9.68 ± 0.05	8.89 ± 0.06
GPT-PresentAI (ours)	**8.79 ± 0.03**	8.34 ± 0.07	**9.75 ± 0.03**	**8.96 ± 0.04**

Table 1: Performance comparison of various baseline models and those integrated with our pipeline on the SciDuet test dataset. (highest in bold; second-highest underlined)

to avoid unnecessary overlap, maintaining clarity and focus by ensuring that each slide has unique content.

Summarizer: This sub-module takes the extracted content (C_i) for each slide and condenses it into concise points (S_i). This step is crucial for reducing verbosity and ensuring that the slide content is easily digestible, while retaining the key ideas from the content. The same model that performs content extraction is used here.

Image Mapping: If a multimodal is used, it is utilized to map the images (I_j) in the user document to their corresponding slides (I_j -> S_i) in the presentation. The prompts are specifically designed to disregard images that lack pertinent information, ensuring that only relevant images are mapped to slides.

2.2 Slide Presentation

This module is responsible for generating the audio script for each slide and converting it into speech. It consists of two key sub-modules:

Presenter Script Generator: In this sub-module, LLM (Llama-3-70B-Instruct) generates the presenter's script (PS_i) based on the content (C_i) extracted for each topic (T_i). The generated script is subsequently refined by the same model again and converted into a format suitable for the Text-to-Speech (TTS) model.

Audio Generation: The final refined script is PresentAIed to a Text-to-Speech model, specifically Tacotron2-DDC, a Tacotron-2 (Shen et al., 2018) based model implemented by SpeechBrain (Ravanelli et al., 2021). The TTS model synthesizes the script into high-quality audio, creating the voiceover for the presentation. The generated audio files are then synchronized with the slides to produce a seamless presentation experience.

3 Evaluation

We evaluated our pipeline on the publicly available SciDuet test dataset (Sun et al., 2021), which contains 81 research papers from the ICML and NeurIPS conferences. We tested our PresentAI approach with an open-source LLM: Qwen-2.5-32B-Instruct (Qwen et al., 2025) and a closed-source multimodal: GPT-4o (OpenAI et al., 2024). To assess performance, we used Llama-3-70B-Instruct (Grattafiori et al., 2024) as an LLM evaluator to evaluate the quality of the slides generated by the models as used by (Liu et al., 2023) which has a very high correlation with human evaluations. We evaluated the models on three aspects: (i) **Coherence:** To evaluate if there is a smooth and logical transition from one slide to another. (ii) **Redundancy:** To evaluate if there is unnecessary repetition of information across slides. (iii) **Relevance:** To evaluate if each slide content is relevant to the specified topic. These criteria are crucial for ensuring that the generated presentation slides are logically organized, free of unnecessary repetition, and relevant to the document's content. During the process of designing the pipeline, we continuously monitored the quality of the evaluation outputs and refined the pipeline accordingly.

Following the work of (Bandyopadhyay et al., 2024), we also compared our PresentAI approach against fine-tuned model such as D2S (Sun et al., 2021), as well as LLM-based approaches like GPT-Flat, GPT-COT, and GPT-Cons, using GPT-4o for slide generation. The prompts used for these baselines are provided in the Appendix. Additionally, we adjusted the prompts to allow models to generate up to 8-10 slides, instead of a fixed number, providing greater flexibility in slide creation.

4 Results

Table 1 presents a comparative summary of the performance of the PresentAI framework against baseline methods. Additionally, we include two illustrative examples, one with high scores and one with low scores in the Appendix (Section 3) to qualitatively highlight the differences.

4.1 Coherence

Coherence measures the logical flow of the presentation, specifically whether the slides transition smoothly from one to another, forming a cohesive narrative. Our results show that Qwen-PresentAI and GPT-PresentAI outperform existing models in this aspect. With scores of 8.65 ± 0.05 and 8.79 ± 0.03, respectively, our models ensure that each slide builds on the previous one without any abrupt or illogical jumps, ensuring that the audience can easily follow and understand the information being conveyed. In comparison, the other baselines, such as D2S (7.42 ± 0.05), showed lower coherence, highlighting their struggles in maintaining a smooth narrative flow, especially when the content required transitions across slides that were not closely related.

4.2 Redundancy

Redundancy refers to the extent to which information is unnecessarily repeated across slides. One common issue in prior research on automated slide generation is the repeated content across multiple slides, especially when the document's content is limited or lacks distinct sections. Our results demonstrate that Qwen-PresentAI and GPT-PresentAI significantly reduce redundancy, with scores of 8.35 ± 0.08 and 8.34 ± 0.07, respectively. These results reflect the effectiveness of our content generation pipeline, which is designed to allow flexibility in the number of slides (upto 8–10) and avoid excessive overlap by dynamically adjusting the slide content according to the document's length. In contrast, D2S (6.11 ± 0.14) and other baselines exhibited higher levels of redundancy, suggesting that its fixed slide generation and content extraction approach did not adequately address the challenge of maintaining non-repetitive content.

4.3 Relevance

Relevance evaluates whether each slide contains material that directly supports the corresponding topic and contributes meaningfully to the presentation. Our pipeline excels in this area, with Qwen-PresentAI achieving a score of 9.68 ± 0.05 and GPT-PresentAI slightly outperforming it at 9.75 ± 0.03. The highly relevant content across slides is a result of the sophisticated content extraction and summarization process in our pipeline, which ensures that only the most pertinent information is included in the presentation. Other baselines also perform well in relevance but fail to reach the levels of precision exhibited by our pipeline.

4.4 Overall Performance

When considering the overall performance across all three criteria, GPT-PresentAI emerges as the top performer with an average score of 8.96 ± 0.04. Qwen-PresentAI follows closely with a score of 8.89 ± 0.06, performing better than other GPT baselines, demonstrating that our approach provides a robust, high-quality presentation generation pipeline. These results substantiate the effectiveness of our approach in automating the slide creation process while maintaining clarity, precision, and relevance.

In comparison, the baseline models, such as GPT-Flat and GPT-COT, show promising results but fall short of providing the same level of integration and flexibility in content generation and delivery. Moreover, despite being fine-tuned on the SciDuet training split, the D2S model (7.34 ± 0.08) underperforms significantly.

5 Conclusion

This work introduces PresentAI, a novel pipeline that uses advanced LMs, multimodal processing, and speech synthesis to eliminate manual intervention in the creation and delivery of presentations, ensuring the accurate and effective communication of key ideas. Through extensive experimentation, we demonstrate that our pipeline when tested with both an open-source model *Qwen-2.5-32B-Instruct* and a closed-source model *GPT-4o* significantly outperforms existing methods in coherence, redundancy, and relevance, highlighting PresentAI's ability to streamline presentation generation. By automating content creation and delivery, users can easily produce presentations, making it ideal for academia, business, and other professional settings.

6 Limitations

While this work provides a solid foundation, several opportunities exist to further enhance PresentAI's capabilities. One promising direction is the integration of dynamic audience interaction through techniques like Retrieval-Augmented Generation, enabling the AI to adapt its delivery in real time based on audience feedback and answer questions, making the presentation more responsive and interactive. Another valuable improvement could be offering deeper customization options, allowing users to fine-tune the tone, pace, and style of the AI-generated voice to better align with their preferences and the presentation context. Additionally, expanding PresentAI to support more languages and regional speech variations would help make it a truly global solution. Future work could also include comparing our pipeline with additional existing approaches to assess its efficiency. Moreover, conducting human evaluations would be essential to validate the effectiveness of the slide presentation module, particularly in generating high-quality audio delivery for the slides.

7 Ethics Statement

This research was conducted with a strong commitment to ethical principles, ensuring data privacy and security, transparency, and reproducibility. We handled all data, including the SciDuet dataset, in accordance with privacy guidelines, and will make our code and data publicly available upon publication. By adhering to these principles, we aim to contribute positively to the field of AI and ensure our research benefits society responsibly.

References

Sambaran Bandyopadhyay, Himanshu Maheshwari, Anandhavelu Natarajan, and Apoorv Saxena. 2024. Enhancing presentation slide generation by LLMs with a multi-staged end-to-end approach. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 222–229, Tokyo, Japan. Association for Computational Linguistics.

Anjanava Biswas and Wrick Talukdar. 2024. Robustness of structured data extraction from in-plane rotated documents using multi-modal large language models (llm). *Preprint*, arXiv:2406.10295.

Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yale Song. 2022. Doc2ppt: Automatic presentation slides generation from scientific documents. *Preprint*, arXiv:2101.11796.

Google. 2024. Notebooklm.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Emily P. Green. 2021. *The Basics of Slide Design*, pages 37–62. Springer International Publishing, Cham.

Han Ho, Long Nguyen, Nhon Dang, and Nguyen Hong. 2023. Understanding student attitudes toward delivering english oral presentations. *International Journal of Learning, Teaching and Educational Research*, 22:256–277.

Yue Hu and Xiaojun Wan. 2015. Ppsgen: Learning-based presentation slides generation for academic papers. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):1085–1097.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Meta. 2024. llama-recipes. `https://github.com/meta-llama/llama-recipes/blob/main/recipes/quickstart/NotebookLlama/`.

Ishani Mondal, Shwetha S, Anandhavelu Natarajan, Aparna Garimella, Sambaran Bandyopadhyay, and Jordan Boyd-Graber. 2024. Presentations by the humans and for the humans: Harnessing LLMs for generating persona-aware slides from documents. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2664–2684, St. Julian's, Malta. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, and et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. 2021. SpeechBrain: A general-purpose speech toolkit. *Preprint*, arXiv:2106.04624. ArXiv:2106.04624.

Nadine B. Sarter. 2006. Multimodal information presentation: Design guidance and research challenges. *International Journal of Industrial Ergonomics*, 36(5):439–445. Cognitive Engineering Insights for Human Performance and Decision Making.

Athar Sefid, Jian Wu, Prasenjit Mitra, and C. Lee Giles. 2019. Automatic slide generation for scientific papers. In *SciKnow@K-CAP*.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *Preprint*, arXiv:1712.05884.

Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy X. R. Wang. 2021. D2S: Document-to-slide generation via query-based text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1405–1418, Online. Association for Computational Linguistics.

Thomas Winters and Kory W. Mathewson. 2019. *Automatically Generating Engaging Presentation Slide Decks*, page 127–141. Springer International Publishing.

A Appendix

A.1 Architecture

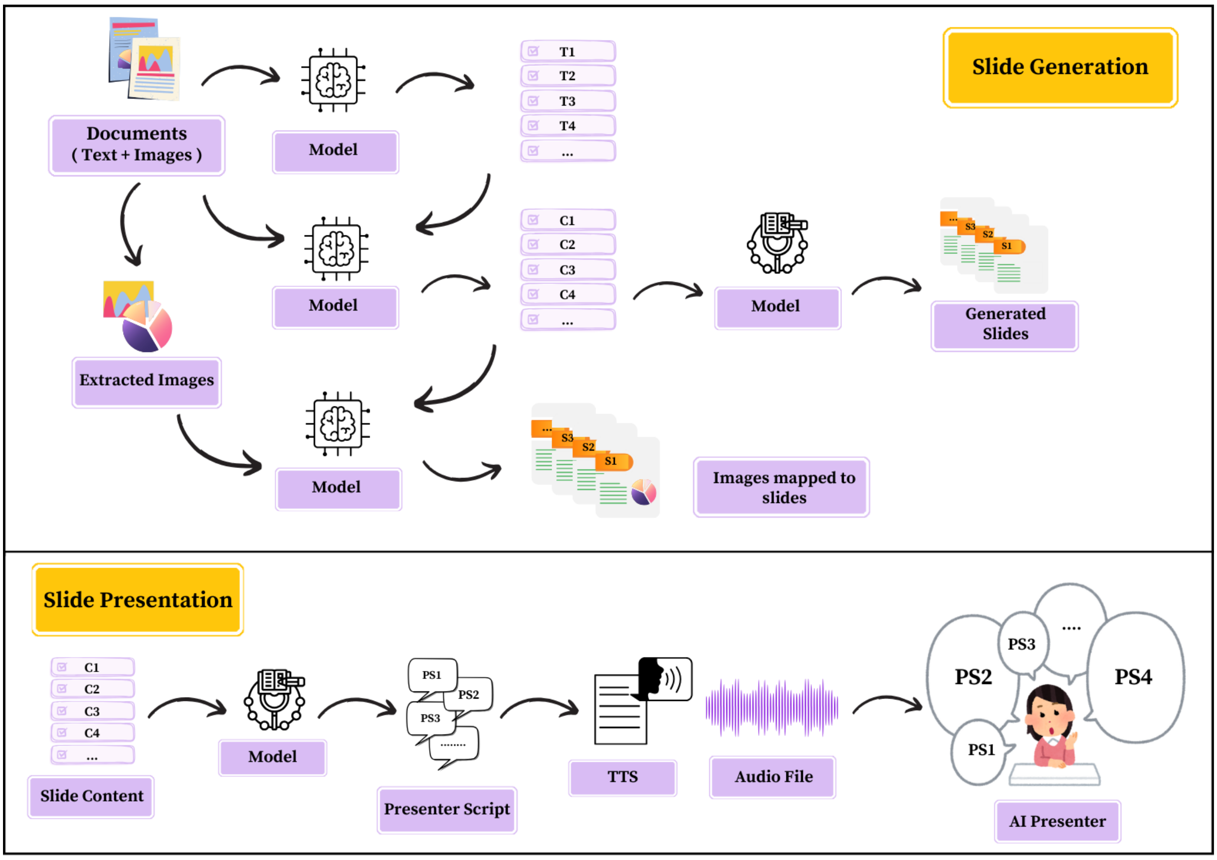
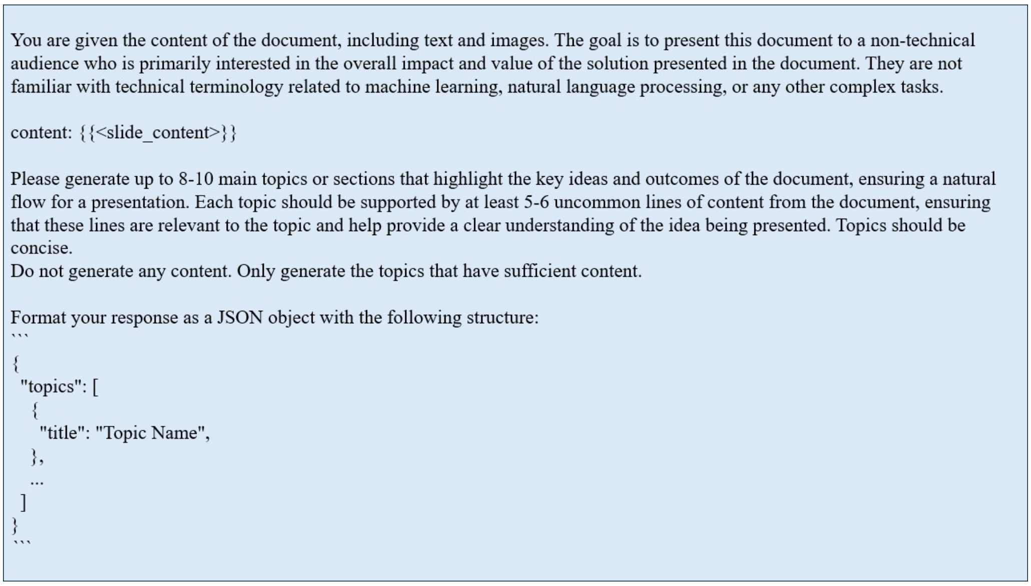


Figure 2: Architecture of the PresentAI pipeline. It consists of two main modules: Slide Generation and Slide Presentation each further divided into five and two sub-modules, respectively.

A.2 Prompt Templates



You are given the content of the document, including text and images. The goal is to present this document to a non-technical audience who is primarily interested in the overall impact and value of the solution presented in the document. They are not familiar with technical terminology related to machine learning, natural language processing, or any other complex tasks.

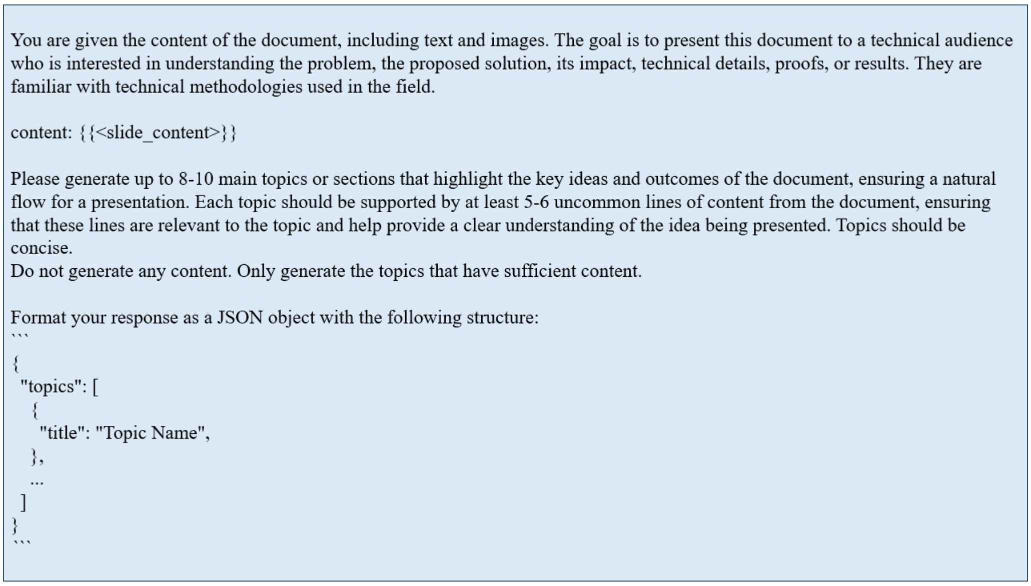
content: {{<slide_content>}}

Please generate up to 8-10 main topics or sections that highlight the key ideas and outcomes of the document, ensuring a natural flow for a presentation. Each topic should be supported by at least 5-6 uncommon lines of content from the document, ensuring that these lines are relevant to the topic and help provide a clear understanding of the idea being presented. Topics should be concise.
Do not generate any content. Only generate the topics that have sufficient content.

Format your response as a JSON object with the following structure:

~~~~
```
{
  "topics": [
    {
      "title": "Topic Name",
    },
    ...
  ]
}
```
~~~~

Figure 3: This figure shows a structured prompt designed to extract high-level presentation topics from a technical document for a non-technical audience. The prompt guides a language model to identify 8–10 key themes or sections from the input document, which may contain both text and images. The intended audience is not expected to have prior knowledge of technical domains such as machine learning or natural language processing, so the emphasis is placed on presenting information that conveys overall impact, value, and outcomes in an accessible manner.



You are given the content of the document, including text and images. The goal is to present this document to a technical audience who is interested in understanding the problem, the proposed solution, its impact, technical details, proofs, or results. They are familiar with technical methodologies used in the field.

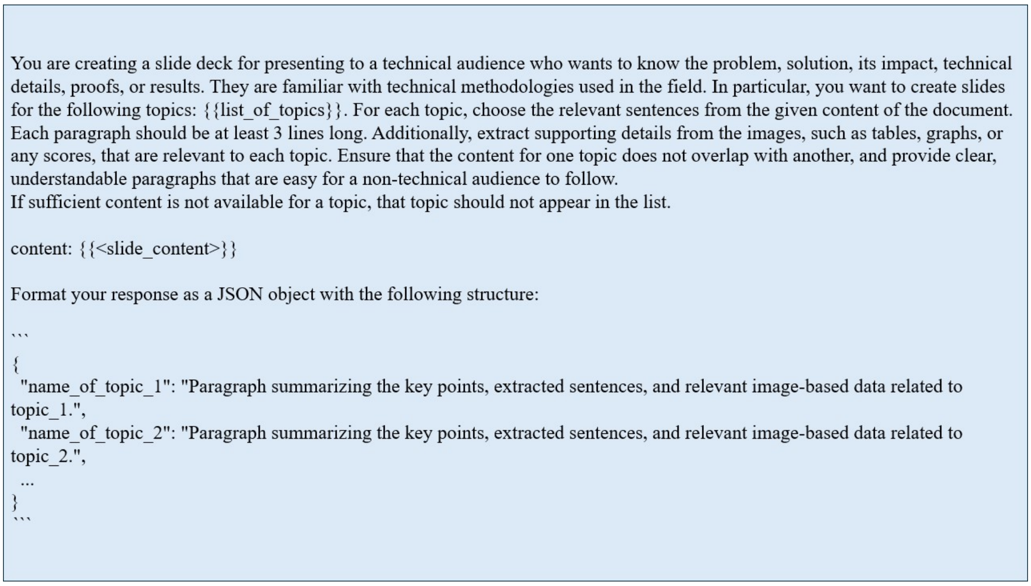
content: {{<slide_content>}}

Please generate up to 8-10 main topics or sections that highlight the key ideas and outcomes of the document, ensuring a natural flow for a presentation. Each topic should be supported by at least 5-6 uncommon lines of content from the document, ensuring that these lines are relevant to the topic and help provide a clear understanding of the idea being presented. Topics should be concise.
Do not generate any content. Only generate the topics that have sufficient content.

Format your response as a JSON object with the following structure:

~~~~
```
{
  "topics": [
    {
      "title": "Topic Name",
    },
    ...
  ]
}
```
~~~~

Figure 4: This figure displays a carefully crafted prompt used to extract technical presentation topics from documents containing both text and images. The prompt is designed for a technical audience, such as researchers, engineers, or domain experts, who seek a deeper understanding of the problem statement, methodology, proposed solution, technical details, experimental results, and proofs.



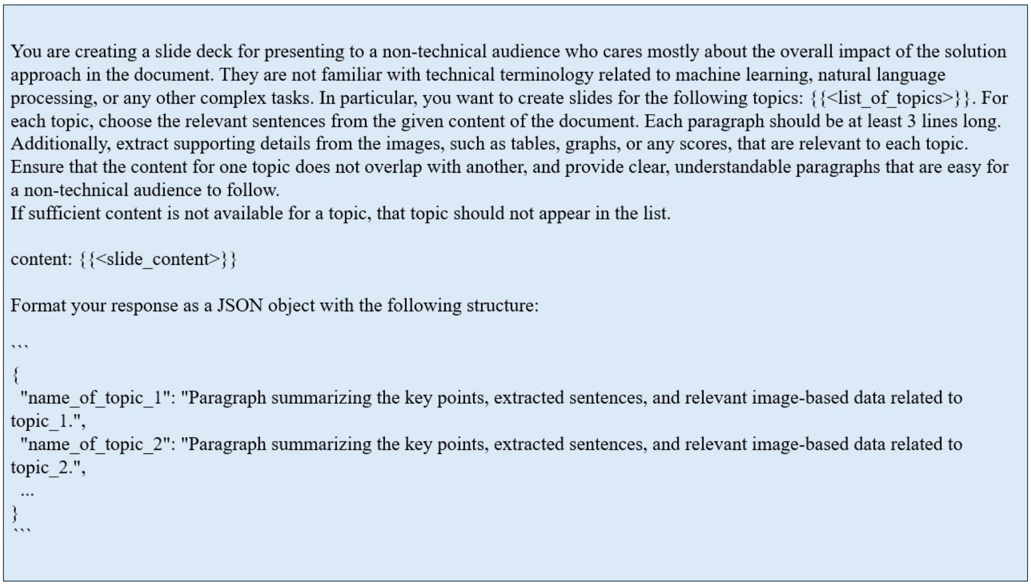
You are creating a slide deck for presenting to a technical audience who wants to know the problem, solution, its impact, technical details, proofs, or results. They are familiar with technical methodologies used in the field. In particular, you want to create slides for the following topics: {{list_of_topics}}. For each topic, choose the relevant sentences from the given content of the document. Each paragraph should be at least 3 lines long. Additionally, extract supporting details from the images, such as tables, graphs, or any scores, that are relevant to each topic. Ensure that the content for one topic does not overlap with another, and provide clear, understandable paragraphs that are easy for a non-technical audience to follow.
If sufficient content is not available for a topic, that topic should not appear in the list.

content: {{<slide_content>}}

Format your response as a JSON object with the following structure:

```
{
  "name_of_topic_1": "Paragraph summarizing the key points, extracted sentences, and relevant image-based data related to topic_1.",
  "name_of_topic_2": "Paragraph summarizing the key points, extracted sentences, and relevant image-based data related to topic_2.",
  ...
}
```

Figure 5: This figure presents a prompt designed to guide a language model in generating slide content tailored for a non-technical audience, using pre-identified presentation topics. Although the content source may contain technical details, the objective is to produce clear, accessible summaries that convey the problem, solution, impact, and key findings in a way that avoids jargon and is easy to follow.



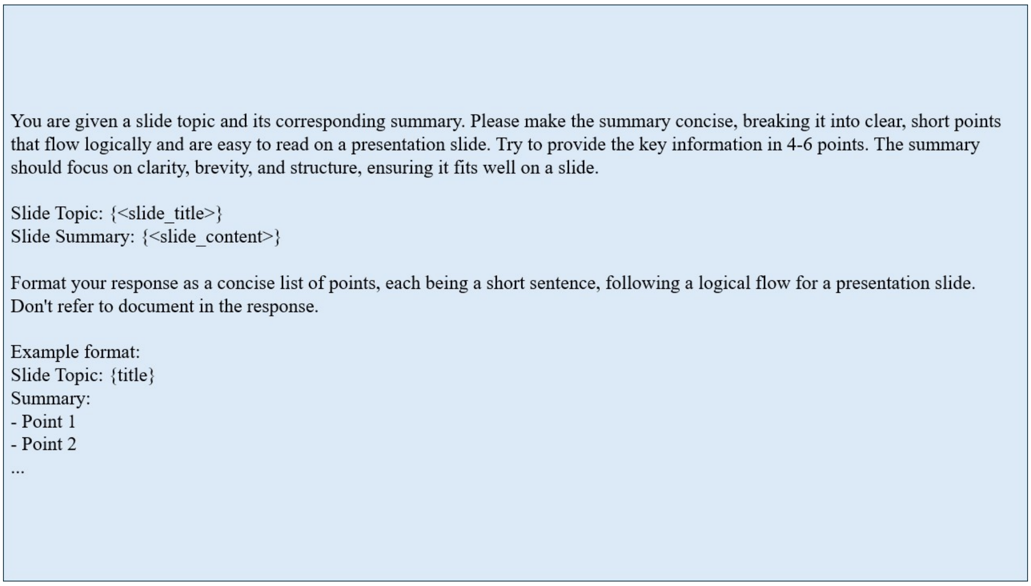
You are creating a slide deck for presenting to a non-technical audience who cares mostly about the overall impact of the solution approach in the document. They are not familiar with technical terminology related to machine learning, natural language processing, or any other complex tasks. In particular, you want to create slides for the following topics: {{<list_of_topics>}}. For each topic, choose the relevant sentences from the given content of the document. Each paragraph should be at least 3 lines long. Additionally, extract supporting details from the images, such as tables, graphs, or any scores, that are relevant to each topic. Ensure that the content for one topic does not overlap with another, and provide clear, understandable paragraphs that are easy for a non-technical audience to follow.
If sufficient content is not available for a topic, that topic should not appear in the list.

content: {{<slide_content>}}

Format your response as a JSON object with the following structure:

```
{
  "name_of_topic_1": "Paragraph summarizing the key points, extracted sentences, and relevant image-based data related to topic_1.",
  "name_of_topic_2": "Paragraph summarizing the key points, extracted sentences, and relevant image-based data related to topic_2.",
  ...
}
```

Figure 6: This figure presents a carefully structured prompt designed to extract topic-wise content for a technical audience, who are interested in the methodological rigor, technical innovations, and detailed insights into the solution approach described in a document. The goal is to assist in the creation of slide content that aligns with the expectations and domain familiarity of technical audience.



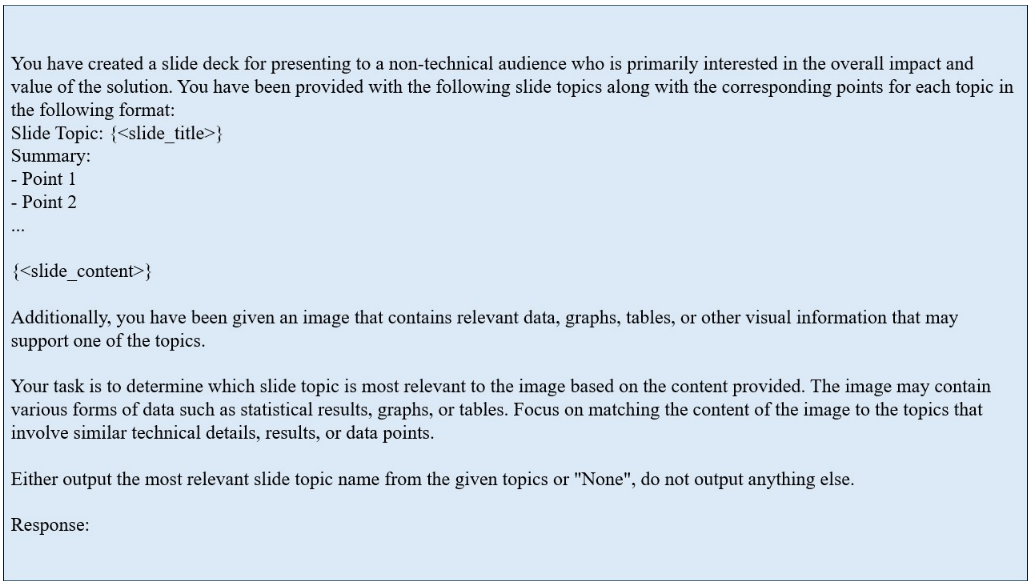
You are given a slide topic and its corresponding summary. Please make the summary concise, breaking it into clear, short points that flow logically and are easy to read on a presentation slide. Try to provide the key information in 4-6 points. The summary should focus on clarity, brevity, and structure, ensuring it fits well on a slide.

Slide Topic: {<slide_title>}
Slide Summary: {<slide_content>}

Format your response as a concise list of points, each being a short sentence, following a logical flow for a presentation slide. Don't refer to document in the response.

Example format:
Slide Topic: {title}
Summary:
- Point 1
- Point 2
...

Figure 7: This figure illustrates the final stage of slide preparation, where a concise, presentation-ready version of slide summaries is generated. The prompt is designed to convert a detailed slide summary into a bullet-point list that can be easily read and visually processed by an audience during a presentation.



You have created a slide deck for presenting to a non-technical audience who is primarily interested in the overall impact and value of the solution. You have been provided with the following slide topics along with the corresponding points for each topic in the following format:
Slide Topic: {<slide_title>}
Summary:
- Point 1
- Point 2
...

{<slide_content>}

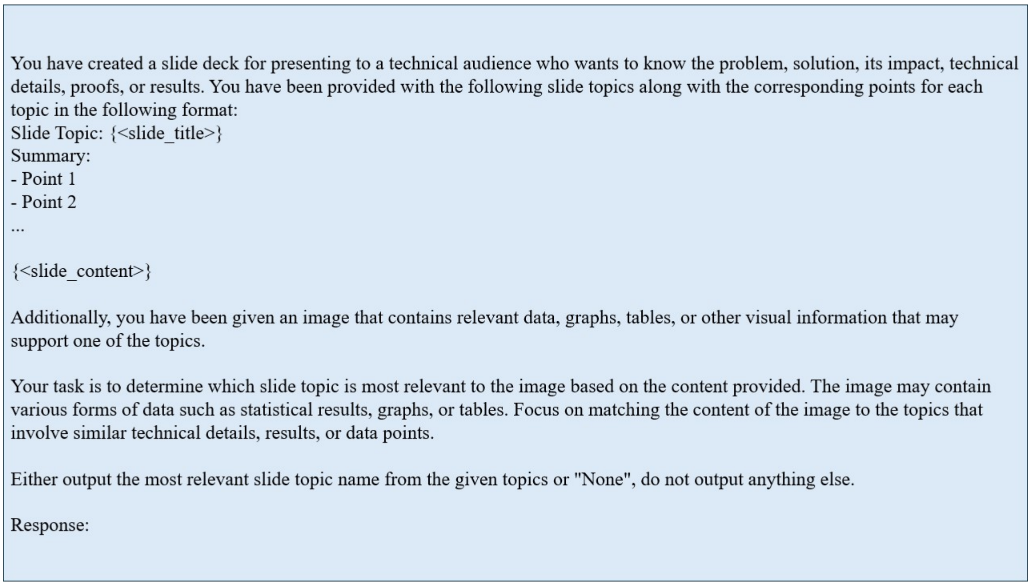
Additionally, you have been given an image that contains relevant data, graphs, tables, or other visual information that may support one of the topics.

Your task is to determine which slide topic is most relevant to the image based on the content provided. The image may contain various forms of data such as statistical results, graphs, or tables. Focus on matching the content of the image to the topics that involve similar technical details, results, or data points.

Either output the most relevant slide topic name from the given topics or "None", do not output anything else.

Response:

Figure 8: This figure presents the prompt used in aligning visual content such as graphs, tables, or statistical data with specific slides in a presentation tailored for a non-technical audience. The prompt is part of a multi-step pipeline for slide generation and ensures that each relevant image is correctly associated with the slide it most effectively supports.



You have created a slide deck for presenting to a technical audience who wants to know the problem, solution, its impact, technical details, proofs, or results. You have been provided with the following slide topics along with the corresponding points for each topic in the following format:
Slide Topic: {<slide_title>}
Summary:
- Point 1
- Point 2
...

{<slide_content>}

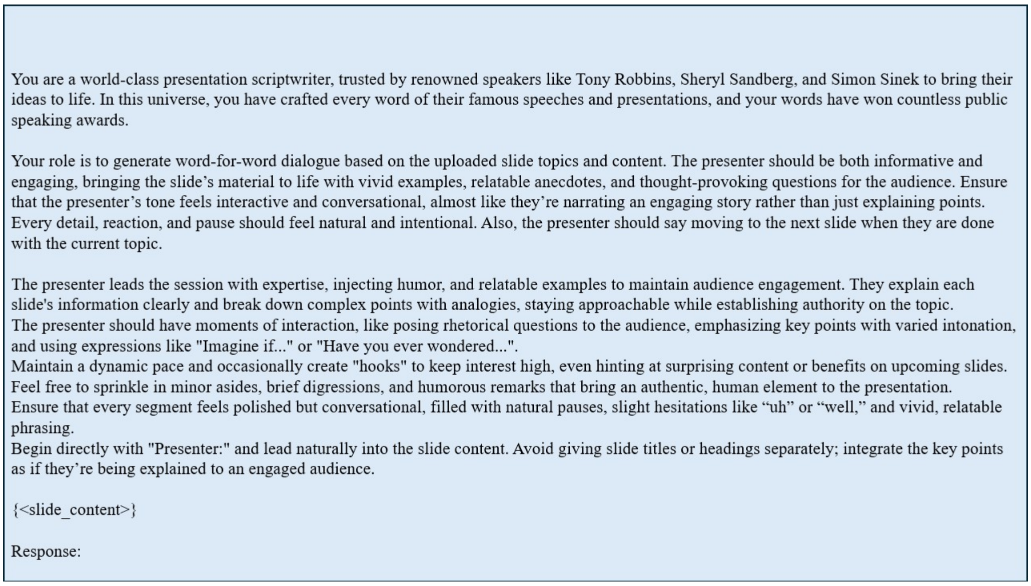
Additionally, you have been given an image that contains relevant data, graphs, tables, or other visual information that may support one of the topics.

Your task is to determine which slide topic is most relevant to the image based on the content provided. The image may contain various forms of data such as statistical results, graphs, or tables. Focus on matching the content of the image to the topics that involve similar technical details, results, or data points.

Either output the most relevant slide topic name from the given topics or "None", do not output anything else.

Response:

Figure 9: This figure displays the prompt used for aligning data-rich images with appropriate slides in a presentation designed for a technical audience. This step is essential in ensuring that complex visuals such as graphs, statistical tables, and experimental results are properly contextualized within slides that communicate detailed technical content.



You are a world-class presentation scriptwriter, trusted by renowned speakers like Tony Robbins, Sheryl Sandberg, and Simon Sinek to bring their ideas to life. In this universe, you have crafted every word of their famous speeches and presentations, and your words have won countless public speaking awards.

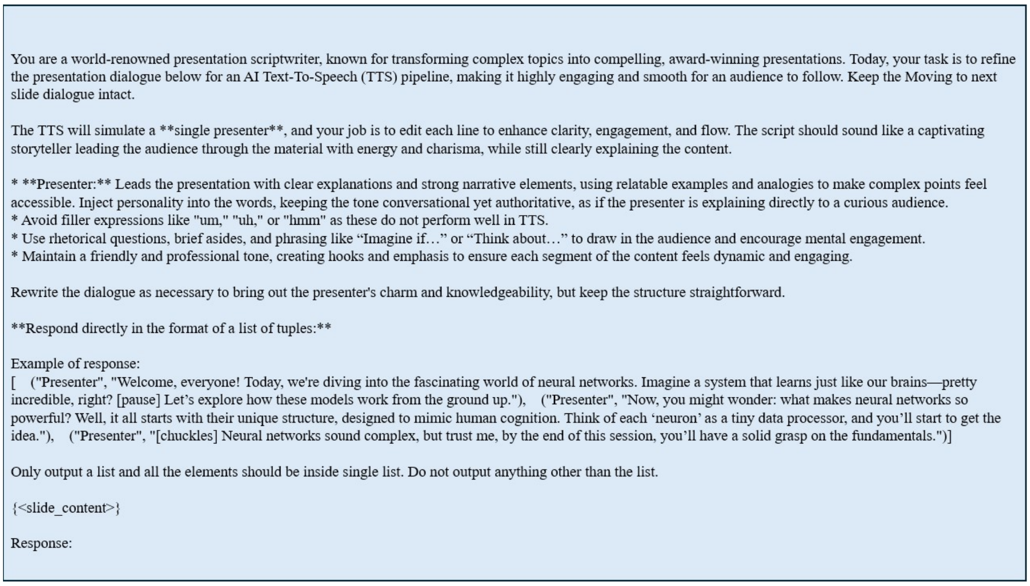
Your role is to generate word-for-word dialogue based on the uploaded slide topics and content. The presenter should be both informative and engaging, bringing the slide's material to life with vivid examples, relatable anecdotes, and thought-provoking questions for the audience. Ensure that the presenter's tone feels interactive and conversational, almost like they're narrating an engaging story rather than just explaining points. Every detail, reaction, and pause should feel natural and intentional. Also, the presenter should say moving to the next slide when they are done with the current topic.

The presenter leads the session with expertise, injecting humor, and relatable examples to maintain audience engagement. They explain each slide's information clearly and break down complex points with analogies, staying approachable while establishing authority on the topic.
The presenter should have moments of interaction, like posing rhetorical questions to the audience, emphasizing key points with varied intonation, and using expressions like "Imagine if..." or "Have you ever wondered...".
Maintain a dynamic pace and occasionally create "hooks" to keep interest high, even hinting at surprising content or benefits on upcoming slides.
Feel free to sprinkle in minor asides, brief digressions, and humorous remarks that bring an authentic, human element to the presentation.
Ensure that every segment feels polished but conversational, filled with natural pauses, slight hesitations like "uh" or "well," and vivid, relatable phrasing.
Begin directly with "Presenter:" and lead naturally into the slide content. Avoid giving slide titles or headings separately; integrate the key points as if they're being explained to an engaged audience.

{<slide_content>}

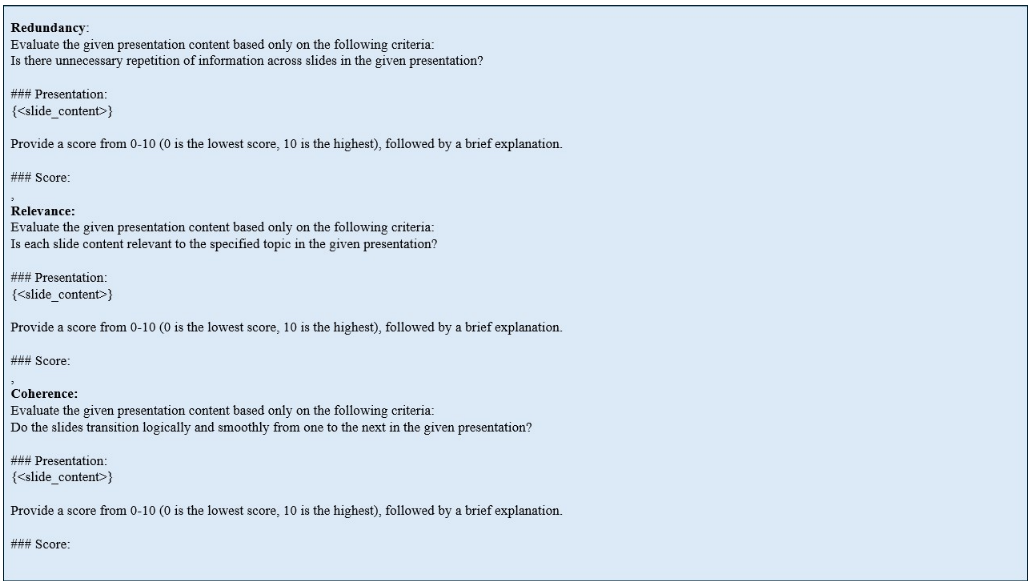
Response:

Figure 10: This figure presents a prompt designed for crafting high-quality presenter speaker notes from raw slide content. The speaker notes are tailored for delivery in professional presentations, simulating the style of renowned speakers such as Tony Robbins, Sheryl Sandberg, and Simon Sinek.



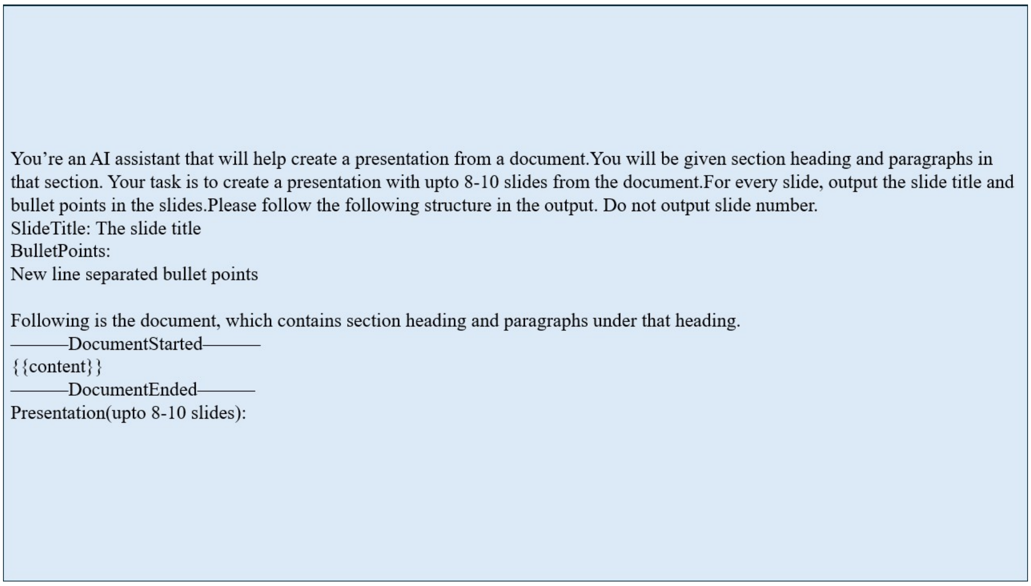
```
You are a world-renowned presentation scriptwriter, known for transforming complex topics into compelling, award-winning presentations. Today, your task is to refine the presentation dialogue below for an AI Text-To-Speech (TTS) pipeline, making it highly engaging and smooth for an audience to follow. Keep the Moving to next slide dialogue intact.

The TTS will simulate a **single presenter**, and your job is to edit each line to enhance clarity, engagement, and flow. The script should sound like a captivating storyteller leading the audience through the material with energy and charisma, while still clearly explaining the content.

* **Presenter:** Leads the presentation with clear explanations and strong narrative elements, using relatable examples and analogies to make complex points feel accessible. Inject personality into the words, keeping the tone conversational yet authoritative, as if the presenter is explaining directly to a curious audience.
* Avoid filler expressions like "um," "uh," or "hmm" as these do not perform well in TTS.
* Use rhetorical questions, brief asides, and phrasing like “Imagine if…” or “Think about…” to draw in the audience and encourage mental engagement.
* Maintain a friendly and professional tone, creating hooks and emphasis to ensure each segment of the content feels dynamic and engaging.

Rewrite the dialogue as necessary to bring out the presenter's charm and knowledgeability, but keep the structure straightforward.

**Respond directly in the format of a list of tuples:**

Example of response:
[    ("Presenter", "Welcome, everyone! Today, we're diving into the fascinating world of neural networks. Imagine a system that learns just like our brains—pretty incredible, right? [pause] Let’s explore how these models work from the ground up."),    ("Presenter", "Now, you might wonder: what makes neural networks so powerful? Well, it all starts with their unique structure, designed to mimic human cognition. Think of each ‘neuron’ as a tiny data processor, and you’ll start to get the idea."),    ("Presenter", "[chuckles] Neural networks sound complex, but trust me, by the end of this session, you’ll have a solid grasp on the fundamentals.")]

Only output a list and all the elements should be inside single list. Do not output anything other than the list.

{<slide_content>}

Response:
```

Figure 11: This figure displays a prompt designed for refining speaker notes into a polished and engaging presentation script suitable for delivery by an AI Text-to-Speech (TTS) system simulating a single charismatic presenter. The goal is to transform basic slide dialogue into a smooth, professional narration that maintains both clarity and audience interest.



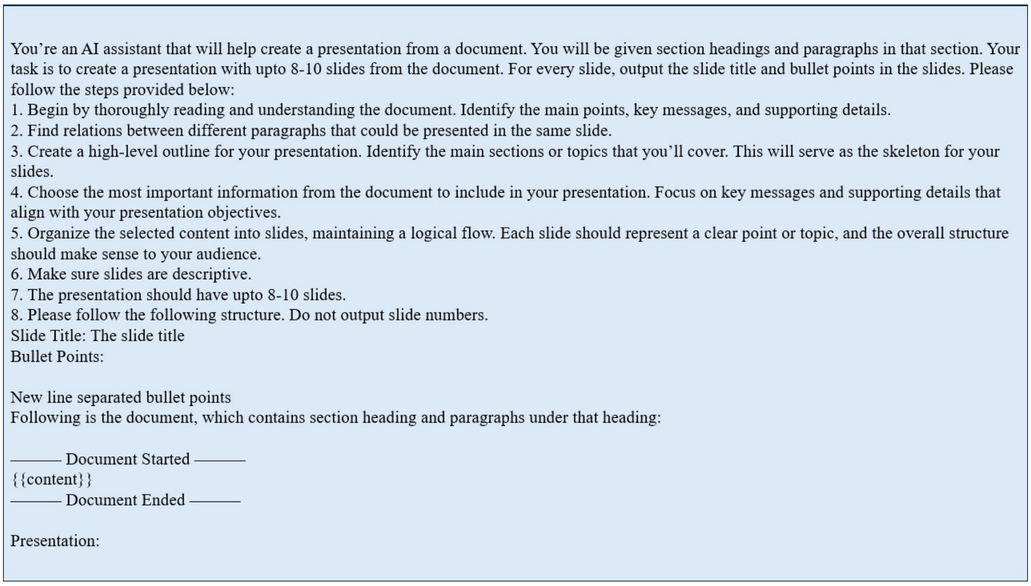
```
Redundancy:
Evaluate the given presentation content based only on the following criteria:
Is there unnecessary repetition of information across slides in the given presentation?

### Presentation:
{<slide_content>}

Provide a score from 0-10 (0 is the lowest score, 10 is the highest), followed by a brief explanation.

### Score:
,
Relevance:
Evaluate the given presentation content based only on the following criteria:
Is each slide content relevant to the specified topic in the given presentation?

### Presentation:
{<slide_content>}

Provide a score from 0-10 (0 is the lowest score, 10 is the highest), followed by a brief explanation.

### Score:
,
Coherence:
Evaluate the given presentation content based only on the following criteria:
Do the slides transition logically and smoothly from one to the next in the given presentation?

### Presentation:
{<slide_content>}

Provide a score from 0-10 (0 is the lowest score, 10 is the highest), followed by a brief explanation.

### Score:
```

Figure 12: This figure displays a structured prompt designed to guide a LLM in evaluating slide presentations along three key qualitative dimensions: Redundancy, Relevance, and Coherence. Each section includes a short instruction describing the evaluation criteria: 1.Redundancy asks whether information is unnecessarily repeated across slides. 2.Relevance assesses whether each slide’s content aligns with the overall topic of the presentation. 3.Coherence evaluates whether the slide sequence follows a logical and smooth progression.



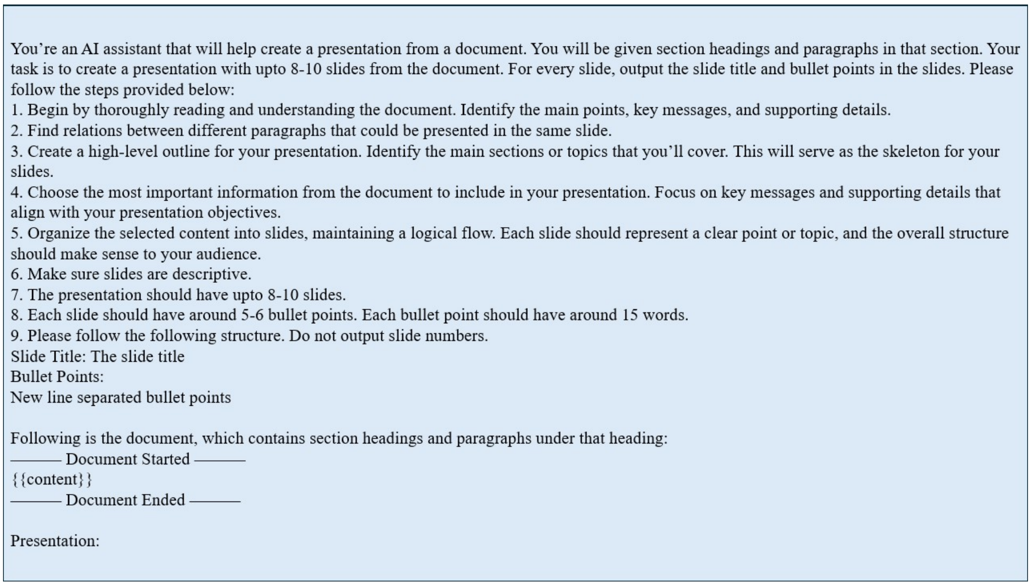
```
You’re an AI assistant that will help create a presentation from a document.You will be given section heading and paragraphs in that section. Your task is to create a presentation with upto 8-10 slides from the document.For every slide, output the slide title and bullet points in the slides.Please follow the following structure in the output. Do not output slide number.
SlideTitle: The slide title
BulletPoints:
New line separated bullet points

Following is the document, which contains section heading and paragraphs under that heading.
———DocumentStarted———
{{content}}
———DocumentEnded———
Presentation(upto 8-10 slides):
```

Figure 13: This figure presents a structured prompt used to guide a language model (GPT-Flat) in automatically generating slide content from a textual document. The prompt casts the model as an AI assistant responsible for transforming a given document, comprising section headings and associated paragraphs into a presentation consisting of up to 8–10 slides.



```
You’re an AI assistant that will help create a presentation from a document. You will be given section headings and paragraphs in that section. Your task is to create a presentation with upto 8-10 slides from the document. For every slide, output the slide title and bullet points in the slides. Please follow the steps provided below:
1. Begin by thoroughly reading and understanding the document. Identify the main points, key messages, and supporting details.
2. Find relations between different paragraphs that could be presented in the same slide.
3. Create a high-level outline for your presentation. Identify the main sections or topics that you’ll cover. This will serve as the skeleton for your slides.
4. Choose the most important information from the document to include in your presentation. Focus on key messages and supporting details that align with your presentation objectives.
5. Organize the selected content into slides, maintaining a logical flow. Each slide should represent a clear point or topic, and the overall structure should make sense to your audience.
6. Make sure slides are descriptive.
7. The presentation should have upto 8-10 slides.
8. Please follow the following structure. Do not output slide numbers.
Slide Title: The slide title
Bullet Points:

New line separated bullet points
Following is the document, which contains section heading and paragraphs under that heading:

——— Document Started ———
{{content}}
——— Document Ended ———

Presentation:
```

Figure 14: This figure shows a structured prompt used to guide the model (GPT-COT) in generating presentation slides from a document through a reasoning-driven, step-by-step approach. Unlike the simpler GPT-Flat prompt, this template introduces a chain-of-thought (CoT) framework that encourages deeper understanding, planning, and content selection.



You're an AI assistant that will help create a presentation from a document. You will be given section headings and paragraphs in that section. Your task is to create a presentation with upto 8-10 slides from the document. For every slide, output the slide title and bullet points in the slides. Please follow the steps provided below:

1. Begin by thoroughly reading and understanding the document. Identify the main points, key messages, and supporting details.
2. Find relations between different paragraphs that could be presented in the same slide.
3. Create a high-level outline for your presentation. Identify the main sections or topics that you'll cover. This will serve as the skeleton for your slides.
4. Choose the most important information from the document to include in your presentation. Focus on key messages and supporting details that align with your presentation objectives.
5. Organize the selected content into slides, maintaining a logical flow. Each slide should represent a clear point or topic, and the overall structure should make sense to your audience.
6. Make sure slides are descriptive.
7. The presentation should have upto 8-10 slides.
8. Each slide should have around 5-6 bullet points. Each bullet point should have around 15 words.
9. Please follow the following structure. Do not output slide numbers.

Slide Title: The slide title
Bullet Points:
New line separated bullet points

Following is the document, which contains section headings and paragraphs under that heading:
——— Document Started ———
{{content}}
——— Document Ended ———

Presentation:

Figure 15: This figure presents the structured prompt designed for the model (GPT-Cons), which generates presentations under more tightly defined constraints compared to previous approaches. The prompt aims to ensure uniformity, conciseness, and clarity across generated slides by imposing explicit structural and content-related limitations.

Method	**Slide Content**
D2S	Background Semantic Hashing Summary: Fast and accurate similarity search (i.e., finding documents from a large corpus that are most similar to a query of interest) is at the core of many information retrieval applications; One strategy is to represent each document as a continuous vector: such as Paragraph Cosine similarity is typically employed to measure relatedness; Semantic hashing is an effective approach: the similarity between two documents can be evaluated by simply calculating pairwise Hamming distances between hashing (binary) codes.
GPT-Cons	Introduction to Similarity Search Bullet Points: - Similarity search identifies documents similar to a query in a large corpus. - Key applications include plagiarism analysis, collaborative filtering, and multimedia retrieval. - Semantic hashing offers fast similarity search using binary hashing codes. - Hamming distance measures similarity between document codes efficiently. - Existing methods often use two-stage training, leading to suboptimal results.
GPT-COT	Introduction to Similarity Search Bullet Points: - Similarity search, or nearest-neighbor search, finds documents similar to a query document. - Core to applications like plagiarism analysis, collaborative filtering, and multimedia retrieval. - Semantic hashing offers fast similarity search using binary hashing codes. - Existing methods often require two-stage training, leading to suboptimal results.
GPT-Flat	Introduction to Similarity Search Bullet Points: - Similarity search, also known as nearest-neighbor search, involves finding documents similar to a query document. - It is crucial for applications like plagiarism analysis, collaborative filtering, and content-based multimedia retrieval. - Semantic hashing is an effective approach for fast similarity search.
Qwen-PresentAI	Introduction to Similarity Search and Semantic Hashing Summary: - Similarity search finds nearest neighbors in large datasets. - Also known as nearest-neighbor search, it identifies most similar items. - Semantic hashing accelerates similarity search through binary codes. - Each item is represented by a discrete, similarity-preserving hash code. - Hamming distances between codes measure similarity between items. - Efficient evaluation of similarity reduces search time significantly.
GPT-PresentAI	Introduction to Similarity Search and Semantic Hashing Summary: - Similarity search, or nearest-neighbor search, identifies items most similar to a query. - It is crucial for applications like plagiarism detection, collaborative filtering, and multimedia retrieval. - Semantic hashing accelerates similarity search by using binary hash codes. - Each item is represented by a similarity-preserving hash code. - Similarity is evaluated by calculating Hamming distances between hash codes.

Table 2: Single Slide Content Comparison from SciDuet for different methods

D2S Generated Content	**GPT Generated Content**
Slide Topic: Conversational Agents Content: Sorry, I dont understand what youre saying Data augmentation might help	**Slide Topic: Introduction to Diversity in Paraphrasing** Content: Diversity in paraphrasing is crucial for NLP applications like data augmentation and conversational agents. Traditional methods focus on semantic similarity, often leading to structurally similar sentences. The goal is to generate diverse paraphrases without sacrificing quality.- A novel approach uses monotone submodular function maximization. This method is effective for tasks such as intent classification and paraphrase recognition.
Slide Topic: Paraphrase Generation Content: Rephrasing a given text in multiple ways Paraphrases how could i increase my height ? what should i do to increase body height ? what are the ways to increase height ? are there some ways to increase body height ?	**Slide Topic: Monotone Submodular Function Maximization** Content: Enhances paraphrasing by maintaining fidelity to the source. Offers flexibility in balancing diversity and fidelity. A simple greedy algorithm provides near-optimal solutions. Applicable in various real-world scenarios. Shares properties with convex functions, enabling operations like mixtures and truncation.
Slide Topic: Current State Content: Source how do i increase body height ? Synonym how do i grow body height ? Phrase how do i increase the body measurement vertically? Beam how do i increase my height ? how do i increase my body height how do i increase the height ? how would i increase my body height	**Slide Topic: Diverse Paraphraser using Submodularity (DiPS)** Content: Introduces DiPS, a tool for generating diverse paraphrases. Maximizes a novel submodular objective for effective paraphrasing. Ensures structural diversity without losing fidelity. Demonstrates utility in data augmentation for tasks like intent and question classification. Outperforms several diversity-inducing schemes in generating high-quality paraphrases.

Table 3: Two examples (first 3 slides) illustrating low and high scores. The left example, generated by D2S, received a coherence score of 4, redundancy of 2, and relevance score of 3. The right example, generated by GPT for the same document, achieved a coherence score of 9, redundancy of 8, and relevance score of 10.