Enhancing Presentation Slide Generation by LLMs with a Multi-Staged End-to-End Approach

Sambaran Bandyopadhyay, Himanshu Maheshwari,
Anandhavelu Natarajan, Apoorv Saxena
Adobe Research
{sambaranb, himahesh, anandvn, apoorvs}@adobe.com

Abstract

Generating presentation slides from a long document with multimodal elements such as text and images is an important task. This is time consuming and needs domain expertise if done manually. Existing approaches for generating a rich presentation from a document are often semi-automatic or only put a flat summary into the slides ignoring the importance of a good narrative. In this paper, we address this research gap by proposing a multi-staged end-to-end model which uses a combination of LLM and VLM. We have experimentally shown that compared to applying LLMs directly with state-of-the-art prompting, our proposed multi-staged solution is better in terms of automated metrics and human evaluation.

1 Introduction

Presentations are a visually effective way to convey an idea to a broad audience (Bartsch and Cobern, 2003). They are heavily used in academia, marketing and sales. A presentation often needs to be generated from a long multimodal document which contains both text and images. A narrative (Castricato et al., 2021) in a presentation generated from a document means (i) the sequence of slide tiles (topics) and (ii) the source content (sections / subsections) from the document for individual slides. Making such a presentation from a document is very time consuming and needs domain expertise.

There are rule-based approaches to generate a presentation from a document (Al Masum et al., 2005; Winters and Mathewson, 2019). Automatically generating a presentation from a given multimodal document is challenging because of several reasons. Compared to a flat document summary, the slide narrative should convey a story to its audience and is often non-linear with respect to the flow of information in the document (Hargood, 2009). The content of a slide should be concise, easy to follow and visually appealing. So, it needs reasoning over both text and images, and their interrelationship. Assuming the slide titles to be the same as the document sections, there are works which use a query specific summarizer Sravanthi et al. (2009), learn sentence importance (Hu and Wan, 2013) and extract hierarchical relations between phrases (Wang et al., 2017) to generate the presentation. Sun et al. (2021) takes the outline from the user and uses that to extract multimodal content and summarize that to slides. Fu et al. (2022) proposes a sequence-to-sequence architecture and a trainable policy to determine when to proceed to the next section/slide. But, it needs large amount of document-to-slides parallel training data which makes it difficult to generalize and scale.

Recent developments in large language models (LLMs) and vision language models (VLMs) have been successfully applied in several multimodal generation tasks. These methods are also easy to use since they can generate content based on simple text prompts and can be generalized to multiple domains. However, compared to open domain generative task, generating a presentation from a specific document is much more challenging because of the following reasons: (i) It is difficult to feed an entire long document to an LLM because of its upper limit on the context length (the number of tokens it can process at a shot) (Mu et al., 2023). (ii) The performance of LLMs drops with the length of the context within a prompt. In fact the performance degrades significantly when LLMs need to access relevant information in the middle of long contexts (Liu et al., 2023a), which is a must requirement for our task. (iii) Finally, LLMs are poor in attributing the exact source of the generated content (i.e., mapping a generated slide to some subsections of the document). Both VLMs and LLMs are prone to hallucinations and this tendency increases with the longer and incomplete context (Azamfirei et al., 2023; Zhou et al., 2023). Thus, directly using

Proceedings of the 17th International Natural Language Generation Conference, pages 222–229
September 23–27, 2024.

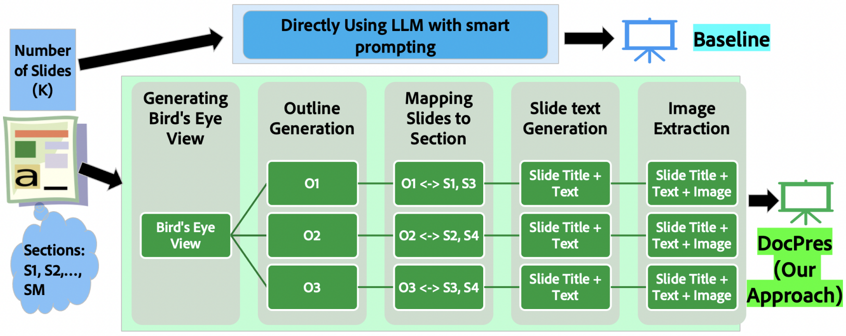

Figure 1: Comparison of DocPres (in green) with a conventional way of generating a presentation directly using an LLM (in blue).

LLMs for generating slides from a long document is not a good strategy.

With this motivation, we try to divide the task of generating presentation from a long document into multiple simpler sub-tasks. The choice of individual smaller tasks is made from three perspectives: (i) How do humans create a presentation from a document? (ii) How to provide only a small amount of context in each call to the LLM? And (iii) How to naturally satisfy properties such as coverage, non-linearity, and source attribution? Following are the *novel contributions* made in this paper: **(i)** We have proposed an unsupervised multi-staged hierarchical approach to generate slides from a long document, referred as DocPres (Document to Presentation). Our approach is multimodal-in and multimodal out in nature. **(ii)** We conduct thorough experimentation involving a state-of-the-art LLM. We are able to show the merit of our multi-staged approach through automated and human evaluations.

2 Proposed Solution Approach

Let the input document be represented as $D = \{(S_i)_{i=1}^{M}, (F_j)_{j=1}^{N})\}$, where S_i is the ith section (or a subsection) and F_j is the jth figure in the document. Both sections and images are associated with their positions (bounding box coordinates and page numbers) in the document. Given the document, we aim to generate a set of slides $L = \{L_1, \cdots, L_K\}$ where each slide has some text and optional images coming from the document.

As the first step, we extract the text and images from the input document (pdf) using a publicly available API [1] which gives the content of the document in a hierarchical fashion with section titles and the corresponding text within them with positions. Images present in the document are also extracted with their locations.

2.1 A Bird's-eye View of the Document

A bird's-eye view of a document refers to its hierarchical summary with sections, sub-sections and content within them. The bird's-eye view is generated as follows: 1. Summarize content in each subsection separately using an LLM. 2. Summarize each section by using an LLM on the text directly under the section and the previously generated summaries of each of its subsections. 3. Combine these summaries along with the hierarchical document structure to obtain the final bird's eye view. This hierarchical approach ensures a layered and comprehensive overview of the document's content.

2.2 Outline Generation

Here, we define the outline of the presentation as the sequence of the slide titles. Outline is important to control the high-level flow of information and convey the story from the document to a broader audience. Feeding the entire document to an LLM has two major drawbacks: limit on the context length of LLMs and their performance drop with the longer context as discussed in Section 1. So, we use the generated bird's-eye view of the document as the context and ask an LLM to generate K important topics with a nice flow and short titles through a chain-of-thought prompt (Wei et al., 2022). The output of the above call is the outline of the presentation as $O = \{O_1, \cdots, O_K\}$, where O_k is the k-th slide title.

2.3 Mapping Slides to Sections

Once we obtain the outline of the presentation, the next task it to generate content for each slide. However, instead of asking the LLM to generate the

[1] https://developer.adobe.com/document-services/apis/pdf-extract/

content directly from the outline and the whole input document as the context, we ask it to associate each slide title to one or more sections of the document using the bird's-eye view of the document as generated in Section 2.1. This has the following advantages: (i) For each generated slide, we can attribute it to some specific sections (and subsections) of the document. This will make the content of the slide more reliable and make it easy for users to update it. (ii) Grounding a slide to some specific parts of the document reduces hallucinations (Yue et al., 2023). (iii) The flow of information in the presentation need not strictly follow the information flow in the given document. This non-linearity of the flow makes the generated presentation more similar to ones prepared by humans (Bartsch and Cobern, 2003). (iv) We do not need to feed the entire document to the LLM, making it suitable for long documents. Appendix has the exact prompt.

Since the output of LLMs are probabilistic in nature and often verbose, we use edit distance (Navarro, 2001) to match each section title produced by the LLM with the ones present in the document. We select the section present in the document when the match is more than 90%. This also makes our system robust to any hallucination in the output produced by the LLM during this mapping.

2.4 Slide Text Generation

Once we get the individual slide titles and the document sections (or subsections) associated to each slide, we target to generate the text content for each slide at a time. If there are multiple sections associated to a slide, we concatenate the content of those sections into a single one before feeding it to the LLM. However, generating the text independently for each slide may not ensure the natural flow of the presentation. Hence, to generate the content of the slide L_k, we feed the Slide Title O_k, concatenated text from the associated sections, along with the slide title and content of the previous slides $L_1, \cdots, L_{k-1}$, $\forall k = 2, \cdots, K$, to an LLM. The detailed prompt is mentioned in Appendix. The output of this stage generates a presentation with the slide titles and the corresponding text in the form of bullet points. We have ensured that the content of each slide is related to its title, maintains a good flow of information and concise in nature.

2.5 Image Extraction

Next, we aim to add images in the slides. We use a set of heuristics and a ranking algorithm based on the similarity of the text and images in a common subspace through a VLM. The content extraction module outputs all the possible images present in a document which can even contain page boundary lines, small and repeated logo, large images with very bad aspect ratio to be shown in a slide, etc. Thus, we use simple rules to remove images with bad aspect ratio and repeated images from the set of candidate images to go into a slide.

Next, for each slide L_k, we use the output of Section 2.3 to get the sections S_{ck} from the document that contributed to the slide. We use the positional information to consider only the figures F_{ck} present within a distance from the contributing sections in the document. After this, a suitability score α_{ck} of each figure F_{ck} is computed as the cosine distance of the CLIP embedding (Radford et al., 2021) of F_{ck} and the CLIP embedding of the text of slide L_K. Then the image with the highest α_{ck} is chosen for the slide L_k subject to $\alpha_{ck} > \alpha_{min}$, where α_{min} is a threshold which we set as 80%.

3 Experimental Evaluation

3.1 Experimental Setup and Baselines

Our proposed approach DocPres does not need any training data since it is based on a combination of pre-trained LLM and VLM (CLIP model). We choose GPT-3.5-turbo (Ouyang et al., 2022) as the LLM, due to its superior performance in many NLP tasks and its larger context length (a requirement by the baselines). We use the publicly available test split of SciDuet dataset (Sun et al., 2021) which consists of 100 research papers from ICML and NeurIPS conferences as our input documents.

We use the following four baselines: (i) **D2S**: We use D2S (Sun et al., 2021) as a semi-automatic baseline where the slide titles are taken from the ground truth slides from SciDuet dataset and the algorithm generated the content of the presentation. (ii) **GPT-Flat**: Here, we feed the entire document to GPT-3.5-turbo and use a descriptive prompt to generate a presentation consisting of slide title and text in bullet points. (iii) **GPT-COT**: Instead of a descriptive prompt, we use chain-of-thought prompting in this baseline with GPT-3.5-turbo. (iv) **GPT-Cons**: We explicitly mention the maximum number of words in a bullet point and the number of bullet point in each slide with COT prompting. The detailed prompts are presented in Appendix.

Method	Coverage (%) ↑		PPL ↓	LLM-Eval ↑
	Paragraph	Sentence		
D2S	38.48 ± 5.43	24.24 ± 3.38	77.38 ± 28.95	7.61 ± 1.05
GPT-Flat	33.41 ± 8.12	22.83 ± 4.03	133.51 ± 96.92	8.94 ± 0.36
GPT-COT	34.83 ± 6.06	23.38 ± 4.07	104.14 ± 53.70	$\mathbf{8.98 \pm 0.26}$
GPT-Cons	34.59 ± 7.63	23.31 ± 4.17	121.37 ± 112.16	8.90 ± 0.33
DocPres	$\mathbf{39.13 \pm 5.68}$	$\mathbf{24.73 \pm 3.48}$	$\mathbf{58.01 \pm 20.44}$	8.95 ± 0.32

Table 1: Results with different automated metrics

3.2 Automated Evaluation Metrics

There is no established evaluation framework exists for document to slides generation. We have carefully chosen three unsupervised metrics here:

1. **Coverage**: It is an unsupervised metric which intuitively capture how much does a subset "cover" the content of the super set. In literature, it has been used for extractive summarization (Kothawade et al., 2020; Jaisankar et al., 2024). We use the following definition of Coverage (at **paragraph** to slide level) in this work:

$$Coverage = \frac{\sum_{\mathbf{e}_p \in D} \sum_{\mathbf{e}_s \in P} cosine(\mathbf{e}_p, \mathbf{e}_s)}{|D||P|} \times 100\%$$

Here, $\mathbf{e}_p$ is a paragraph embedding from the given document and $\mathbf{e}_s$ is a slide embedding from the generated presentation as obtained by a sentence transformer model (Reimers and Gurevych, 2019). Similarly, coverage can also be computed ta **sentence** level by replacing a paragraph with a sentence from the document and a slide with a bullet point (or sentence) from the presentation in the equation above. Sentence level coverage offers a finer granularity than paragraph-level coverage. More is the Coverage, better is the presentation.

2. **Perplexity (PPL)**: Perplexity is a metric to indicate the fluency of the generated text. It is obtained using GPT-2, as discussed in Liu et al. (2021). Perplexity measures how likely the language model (GPT-2 here) is to generate the sequence. If GPT-2 assigns a high probability to the token present in the sentence, the perplexity will be lower, indicating a fluent and grammatically correct sentence.

3. **LLM-Eval for presentation quality**: G-Eval (Liu et al., 2023b) is a well-established metric that uses GPT to evaluate various NLP tasks. It has a very high correlation with humans. We believe that G-Eval might be biased to GPT output, so instead of GPT, we use open-source LLMs (Mistral-7B-Instruct-v0.2). We call this metric LLM-Eval. We use LLM-Eval to measure the overall presentation quality in terms of organization, effectiveness, clarity, coherence, and the ability to convey complex ideas to the audience.

3.3 Results and Analysis

Table 1 compares the performance of DocPres with the baselines. Please note that D2S has some advantage on SciDuet dataset since it was specifically trained on the same dataset where all other algorithms including DocPres are LLM-based. Interestingly, DocPres performs the best among the baselines for Coverage and PPL, where the margin is significant compared to other LLM based approaches. For LLM-Eval, all the LLM-based approaches perform very close to each other. This specifically supports our hypothesis that dividing a complex task into smaller sub-tasks and providing limited context for each sub-task helps to improve the overall performance of an LLM.

3.4 Human Evaluation

We have also conducted a small scale human survey to understand the quality of the generated presentation to human experts. First, we selected five research papers from ACL workshops which are relatively easy to follow. We hired [2] two professional reviewers who have reasonable understanding of NLP and have good presentation generation skill. Based on our discussion with subject matter experts, we decided the following criteria to evaluate the quality of a generated presentation from a given document: (1) Readability: *How good is the language and readability?*, (2) Consistency: *Is a slide title consistent with the slide content?*, (3) Coverage: *Does the presentation cover all important parts of the document?*, (4) Diversity: *Is the content of the presentation non-repetitive enough?*, (5) Flow: *How is the flow of information in the presentation?* and (6) Usability: *Is the generated presentation good enough for an initial draft?*. The

[2] https://www.upwork.com/

Method	Readability	Consistency	Coverage	Diversity	Flow	Usability
GPT-Flat	2.30 ± 1.16	2.20 ± 1.13	1.30 ± 0.48	2.80 ± 1.54	1.70 ± 0.67	1.20 ± 0.63
GPT-COT	2.30 ± 1.16	2.40 ± 1.35	1.50 ± 0.85	2.80 ± 1.54	1.70 ± 0.67	1.20 ± 0.63
GPT-Cons	2.30 ± 1.16	2.00 ± 1.05	1.10 ± 0.31	2.80 ± 1.54	1.70 ± 0.67	1.20 ± 0.63
DocPres	**3.90 ± 0.73**	**3.80 ± 1.39**	**2.70 ± 1.16**	**2.90 ± 1.44**	**2.70 ± 0.82**	**3.20 ± 1.22**

Table 2: Results of human evaluation

evaluators are instructed to score a generated presentation against each of these metrics in a scale of 1 (lowest in quality) to 5 (best in quality) [3].

Human evaluation results in Table 2 shows that the slides generated by DocPres are consistently rated high by human experts with a good margin compared to the baselines. Interestingly, the scores of all direct GPT-based baselines are very close to each other showing that different prompting techniques could not generate visible difference in the generated presentation. The reviewer appreciated the output of DocPres from different perspectives such as *"The language and grammar are all fine"*, *"The main text and the slide title are closely related"*, *"The flow is good"*, etc. However, there were a few concerns such as DocPres *"Covers a lot of content from the PDF but does not deep dive"* and *"The deck keeps on repeating the benefits of text mining"*. We also asked reviewers to comment on the images extracted by DocPres. Reviewers consistently appreciated the precision of the selected images (because of our filtering strategy), however complained about the missing images. This is because research papers have many non-natural images which CLIP based algorithm fails to understand. Overall, the reviewers agree that compared to the baselines, the generated presentations from DocPres can serve well as an initial draft.

4 Discussions and Conclusion

This work presented a novel multi-staged framework for generating presentations from documents. By breaking down the task into five sub-tasks, our approach achieved significant improvements compared to baselines including single-shot prompting to LLMs. Comprehensive evaluations, both automatic and human, confirmed the merit of our multi-stage approach. The presentations from our approach demonstrated better coverage, readability, consistency, diversity, flow, and overall usability. The success of our multi-stage approach highlights the benefits of decomposing complex tasks into smaller and well-defined subtasks, with limited context, for LLMs.

Limitations

There are some limitations of our current work. First, our image selection approach is constrained by CLIP's limitations. Since CLIP is trained on datasets mainly consisting of naturally occurring items like photographs and cartoons, it struggles with document images such as illustrations, flowcharts, and graphs. Next, although we have not yet analyzed the computational cost of our methodology, we believe there is potential for cost reduction, as it heavily relies on LLM usage. Finally, our method currently converts a single document into a presentation, which is suitable for many academic presentations. However, it does not address scenarios where information from multiple documents needs to be combined into a single presentation.

Acknowledgments

We extend our gratitude to our Adobe Research colleagues and INLG 2024 reviewers for their valuable insights and feedback on this work.

References

S.M. Al Masum, M. Ishizuka, and M.T. Islam. 2005. 'auto-presentation': a multi-agent system for building automatic multi-modal presentation of a topic from world wide web information. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 246–249.

Razvan Azamfirei, Sapna R Kudchadkar, and James Fackler. 2023. Large language models and the perils of their hallucinations. *Critical Care*, 27(1):1–2.

Robert A Bartsch and Kristi M Cobern. 2003. Effectiveness of powerpoint presentations in lectures. *Computers & education*, 41(1):77–86.

Louis Castricato, Stella Biderman, David Thue, and Rogelio Cardona-Rivera. 2021. Towards a model-theoretic view of narratives. In *Proceedings of the*

[3] We could not use D2S here since we were not able to run its available code on any other dataset except SciDuet.

Third Workshop on Narrative Understanding, pages 95–104.

Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yale Song. 2022. Doc2ppt: automatic presentation slides generation from scientific documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 634–642.

Charlie Hargood. 2009. *Exploring the Importance of Themes in Narrative Systems*. Ph.D. thesis, University of Southampton.

Yue Hu and Xiaojun Wan. 2013. Ppsgen: learning to generate presentation slides for academic papers. In *Twenty-Third International Joint Conference on Artificial Intelligence*. Citeseer.

Vijay Jaisankar, Sambaran Bandyopadhyay, Kalp Vyas, Varre Chaitanya, and Shwetha Somasundaram. 2024. Postdoc: Generating poster from a long multimodal document using deep submodular optimization. *arXiv preprint arXiv:2405.20213*.

Suraj Kothawade, Jiten Girdhar, Chandrashekhar Lavania, and Rishabh Iyer. 2020. Deep submodular networks for extractive data summarization. *arXiv preprint arXiv:2010.08593*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. 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.

Yixin Liu, Graham Neubig, and John Wieting. 2021. On learning text style transfer with direct rewards. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4262–4273, Online. Association for Computational Linguistics.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens. *arXiv preprint arXiv:2304.08467*.

Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

M Sravanthi, C Ravindranath Chowdary, and P Sreenivasa Kumar. 2009. Slidesgen: Automatic generation of presentation slides for a technical paper using summarization. In *Twenty-Second International FLAIRS Conference*.

Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy XR 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.

Sida Wang, Xiaojun Wan, and Shikang Du. 2017. Phrase-based presentation slides generation for academic papers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Thomas Winters and K. Mathewson. 2019. Automatically generating engaging presentation slide decks. In *EvoMUSART*.

Xiang Yue, Boshi Wang, Kai Zhang, Ziru Chen, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. *arXiv preprint arXiv:2305.06311*.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2023. Analyzing and mitigating object hallucination in large vision-language models. *arXiv preprint arXiv:2310.00754*.

```
From the following text which contains a set of headings and some
content within each heading:

TEXT

Extract the most important headings present in it.
Reduce the length of each heading to five words if they are lengthy.
```

Table 3: Prompt to generate an outline.

```
Think step by step

You are given with the following title:
{outline_headings}

and a list of keys:
{document_heading_from_bird_eye_view}

Each key is associated with some text as presented in the dictionary format
below:
{bird_eye_view}

The task is to find 1-2 significantly matched keys. The matching should be
done based on the similarity of the text associated with the keys with the
given heading.
Matching keys are: <semicolon separated list if more than a single key>
```

Table 4: Prompt to map slides to section.

Appendix

A Prompt to Generate an Outline

Table 3 shows the prompt that we used for generating the outline of the presentation.

B Prompt to Map Slides to Sections

Table 4 shows the prompt that we used for mapping slides to sections.

C Prompt to Generate Slide Content

Table 5 shows the prompt that we used for generating the slide content.

```
You are a presentation generator from a source of text. You have to generate
the slide number {slide_index}.
Previous slide headings and slide contents are given below in the format of a
list of dictionaries.
{previous_slide}
Given the following slide heading and the source of text respectively, create
the content of the slide number {slide_index} such that:
1. The slide should have maximum max_bullet bullet points.
2. Ensure that the content of the bullet points are coming strictly from the
given source of text only.
3. The content of the slide is very relevant to the given slide heading
4. Each bullet point should have a maximum of 10 words
5. Ensure that this slide does not have any content repeated from the
previous slides.
6. The flow of the overall presentation is nice.
7. Do not prefix the slide title before the bullet points in the output

Slide Title: HEADING

Source of text: TEXT
```

Table 5: Prompt to generate slide.

```
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 ONLY ##number_of_slides## 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.
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———-
##document##
———-Document Ended———-

Presentation (only ##number_of_slides## slides):
```

Table 6: Prompt for GPT-Flat

```
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 ONLY ##number_of_slides## 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. Presentation should have only ##number_of_slides## slides.
8. Please follow the following structure. Do not output slide number.
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———-
##document##
———-Document Ended———-

Presentation:
```

Table 7: Prompt for GPT-COT.

D Prompt for the Baselines

D.1 Prompt for GPT-Flat

Table 6 shows the prompt for GPT-Flat baseline.

D.2 Prompt for GPT-COT

Table 7 shows the prompt for GPT-COT baseline.

D.3 Prompt for GPT-Cons

Table 8 shows the prompt for GPT-Cons baseline.

D.4 Prompt for LLM-EVal

Table 9 shows the prompt we used for LLM-Eval to evaluate the presentation quality.

```
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 ONLY ##number_of_slides## 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. Presentation should have only ##number_of_slides## slides.
8. Each slide should have around 7 bullet points. Each bullet point should
have around 15 words.
9. Please follow the following structure. Do not output slide number.
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————-
##document##
————-Document Ended————-

Presentation:
```

Table 8: Prompt for GPT-Cons

```
On a scale of 0-10, rate the effectiveness, clarity, and overall quality of
the following text presentation, considering factors such as organization,
coherence, and the ability to convey complex ideas to the audience.
0 is the lowest score, whereas 10 is the highest score.

Presentation:
##presentation##

Score (an integer between 0 and 10):
```

Table 9: Prompt for LLM-Eval to evaluate the final presentation quality.