Deep Submodular Optimization and LLM for Multimodal Content Extraction and Automatic Poster Generation from Long Document

Vijay Jaisankar[1]*, Sambaran Bandyopadhyay[2], Kalp Vyas[3], Varre Suman Chaitanya[3], Shwetha Somasundaram[2]

[1]International Institute of Information Technology, Bangalore
[2]Adobe Research
[3]IIT Bombay
{vijayjaisankar.vj, samb.bandyo, kalp.s.vyas}@gmail.com, 200050153@iitb.ac.in, shsomasu@adobe.com

Abstract

A poster from a long input document can be considered as a one-page easy-to-read multimodal (text and images) summary presented on a nice template with good design elements. Automatic transformation of a long document into a poster is a very less studied but challenging task. It involves content summarization of the input document followed by template generation and harmonization. In this work, we propose a novel deep submodular function which can be trained on ground truth summaries to extract multimodal content from the document and explicitly ensures good coverage, diversity and alignment of text and images. Then, we use an LLM based paraphraser and propose to generate a template with various design aspects conditioned on the input content. We show the merits of our approach through extensive automated and human evaluations.

1 Introduction

Recent success of large language models (OpenAI 2023; Petroni et al. 2020) and large vision language models (Radford et al. 2021; Li et al. 2022) have led to several new applications in the field of Generative AI. In this work, we focus on transforming a long multimodal document, containing both text and images, into a poster - which is a visually rich one-page multimodal summary of the document. A poster should have good coverage of the overall content of the document and is generally easy-to-read and presented in a nice template with good design elements (Shelledy 2004). To create a poster, there are different types of design tools and products available such as Microsoft PowerPoint and Adobe Express. However, using them requires substantial manual effort to select multimodal content from a document and determine the suitable design elements for the poster. It is often time-consuming and requires domain expertise.

Automatic transformation of a document into a poster is a relatively less studied problem (Qiang et al. 2019; Xu and Wan 2021, 2022). Such a transformation process mainly involves two major steps: (i) Content summarization (or planning), which aims to select key content from the document and paraphrase them in some appropriate format to be put in the poster and (ii) Template generation and harmonization, which involves generating a suitable layout and design elements of the poster such as background, font and size of letters, number of text and image elements etc. and finally filling up the generated template with the generated content. Content summarization for poster is challenging because of the following reasons: A poster is very limited in size (single large page), but the input document can have multiple pages. Thus, it is essential that the content in the poster (i) represents all the important aspects of the input document (coverage), (ii) has very less repetition within it (diversity) and (iii) has well aligned images and text. Coverage and diversity have been studied in text summarization literature (Lin and Bilmes 2011). But their interpretation in the multimodal setup (with text and images) is not well understood.

Recent advent of zero-shot and few-shot LLMs have significantly improved the state-of-the-art (SOTA) for several natural language processing tasks like summarization (OpenAI 2022, 2023). However, using them directly for content summarization in poster generation is difficult because: (i) Many existing LLMs are text based. The ability of vision or multimodal language models for long document summarization is still questionable (Islam and Moushi 2024; Meng et al. 2024). (ii) LLMs tend to hallucinate and their performance drops when the input context is very long (Liu et al. 2023). However, documents can be lengthy and information in any parts of the document can be important for the poster. (iii) LLM fine-tuning is possible. But it needs expensive hardware support and good amount of data (Han et al. 2024). Whereas, use of LLM as a service is computationally less demanding but does not suit the purpose if used directly (Sun et al. 2022). (iv) There are generic Retrieval Augmented Generation (RAG) techniques available in the literature (Gao et al. 2023). But for poster generation, we need to explicitly assure that the retrieved multimodal content has all the intrinsic properties mentioned above.

In this paper, we have addressed all the above challenges by proposing an approach to handle multimodal content jointly and avoid feeding the entire content to an LLM directly. These are the key contributions made in this work: **(1)** We propose an efficient and computationally fast end-to-end pipeline, referred as *PostDoc* (in Figure 1), for automatically generating a visually rich *Post*er from a long multimodal input *Doc*ument. **(2)** For selecting suitable content from the

*This work was done when Vijay, Kalp and Varre were interns at Adobe Research

input document, we propose a novel optimization formulation using deep submodular functions which explicitly ensures coverage, diversity and multimodal content alignment in the multimodal summary, and trainable on ground truth data. This summary is passed to an LLM (GPT-3.5-turbo) for paraphrasing content to be put into a poster. **(3)** We conduct thorough experimentation to validate the quality of the content and design aspects of the generated posters through automated and a small-scale human evaluations.

2 Related Work and Background

Multimodal Summarization: Text summarization has been studied quite extensively in the literature (Zhang et al. 2020; Zhong et al. 2020). From the last few years, researchers have focused on multimodal summarization which involves different modalities such as text, images and videos, and leverages cross-modal information (Mademlis et al. 2016; Li et al. 2017). Existing approaches for multimodal summarization suffers from modality-bias problem (Zhu et al. 2018), generating summaries with limited number of images (Zhu et al. 2020) and treating text and images as different entities (Zhang et al. 2022). There are approaches which deals with text and videos (He et al. 2023), and use additional information such as a graph (Zhang et al. 2021). Submodular functions have been used for multiple summarization tasks (Lin and Bilmes 2011; Tschiatschek et al. 2014; Modani et al. 2016; Zhu et al. 2018) as they are a natural fit for text summarization. A set of differentiable submodular functions, also known as deep submodular functions (Bilmes and Bai 2017; Kothawade et al. 2020) have been proposed in the literature and also used for text summarization. However, they have not yet incorporated multimodal aspects such as multimodal coverage, diversity, and image-text alignment terms which are key components for multimodal summarization. In this work, we address this research gap and aim to design a deep submodular function which captures a set of intrinsic properties of multimodal extractive summarization and is also trainable from the ground truth data.

Layout Generation: State-of-the-art research works on template/layout generation can be of two categories: (i) Diffusion based techniques such as LayoutDM (Chai, Zhuang, and Yan 2023) and (ii) LLM based techniques such as LayoutPrompter (Lin et al. 2024). Diffusion based models for layout generation are typically capable of generating creative layouts with design elements. However, they are very prone to failures with generated layouts that have several overlapping or unaligned bounding boxes. LLM based models tend to generate a text based plan containing the details of the shape and size of the bounding boxes, color code, etc. of the layout in text. Since LLMs are still not good enough in mathematical reasoning, the generated layouts often have the problems with less contrastive colors and overlapping or unaligned bounding boxes. Moreover, these approaches do not consider the actual content when generating the layout. So, they cannot be used directly to generate a poster layout.

Document Transformation: There are few existing works related to poster creation from documents. Xu and Wan (2021) generates posters from documents but relies on template retrieval from a fixed set of templates and also limiting its applicability to research papers exclusively. There are few research works on automatically generating slides from scientific documents (Sun et al. 2021; Fu et al. 2022; Maheshwari et al. 2024), but they often need users to come up with an outline specific to slide presentation or summarize individual sections in each slide.

2.1 Background on Submodular Functions

Here, we discuss some key concepts required to understand our solution. A submodular function f is a set function with diminishing returns property. In simple terms, adding an element to a smaller set provides a larger marginal gain compared to adding the same element to a larger set. Mathematically, for sets $A \subseteq B$, on adding extra element $x \notin B$, $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$ (Lin and Bilmes 2011). A key advantage of using a submodular function is that there exists a simple greedy algorithm of iteratively choosing the element that maximises the marginal gain with an approximation guarantee.

Recently, researchers explore deep (trainable) submodular functions (Bilmes and Bai 2017; Kothawade et al. 2020) where the data is projected into a suitable feature or embedding space. They can be represented as: $f(A) = \sum_{u \in \mathbb{U}} \Phi\Big(w(u) m_u(A)\Big)$, where Φ is a non decreasing non-negative concave function, $w(u)$ represents the trainable weight of the feature u, $m_u(S) = \sum_{s \in S} m_u(s)$ is a non-negative modular function. These functions enable the learning of submodular functions through a training dataset and also their inference is fast compared to the quadratic complexity of most of the other submodular functions.

3 Problem Statement and Solution Approach

As discussed in Section 1, we aim to automatically generate a poster from a long document. The document may contain different types of multimodal content such as text, images, tables, charts, etc. For the ease of presentation, we use image to represent all such non-textual elements in a document. As shown in Figure 1, we use the Adobe Extract API [1] to extract the multimodal content from the document. Then, we use the pre-trained multimodal model BLIP (Li et al. 2022) to encode both text and image elements into a common vector space of dimension 768. We have observed that the embeddings of text and images are often not in the same scale. To overcome this issue, we first shift all the embeddings to positive coordinate of the embedding space and do an L1 normalization of the embeddings.

With this, we present the problem mathematically as follows. Given $D = (e_1, e_2, \cdots, e_N)$ is a multimodal input document with N (can vary over the documents) content elements in order where each content element can be a text sentence or an image. We assume that each $e_i \in \mathbb{R}^d_+$ denotes a d dimensional normalized BLIP embedding (as discussed above) of the ith content element in the document ($d = 768$ in this case). Our goal is to select a subset of

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

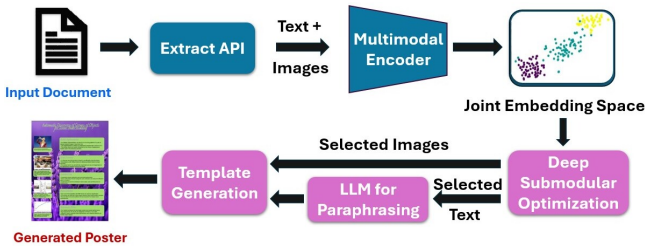

Figure 1: Block Diagram of PostDoc

content elements $A \subseteq D$ with $|A| \leq K$ so that content elements of A have good coverage, diversity and alignment as discussed in Section 1. Since the size of the poster is limited, we extract the corresponding maximum $K > 0$ content elements from the document. We discuss how to fix K in Section 4. For training our content selection algorithm, we use a training set of documents with their ground truth summary. So, we assume to have the following training set $\mathcal{T} = \{(D_1, A_1^*), (D_1, A_1^*), \cdots, (D_M, A_M^*)\}$ containing M pairs of documents and the corresponding ground truth summary. During the inference for a given document D, we will first select the subset A using the trained content selection model. Then we will paraphrase A into a poster friendly format. We also generate a poster template with suitable design elements based on the content and put the paraphrased content into it.

3.1 Multimodal Extractive Summarization

Once the embeddings of all the content elements are obtained from a document, the next step is to select only a subset of them such that desired properties are satisfied. We propose a novel deep submodular based optimization framework for this task. As mentioned in Section 3, the normalized BLIP embeddings of the sequence of content elements (text sentences or images) from the document are represented as $D = (e_1, e_2, \cdots, e_N)$, with $e_i \in \mathbb{R}_+^d$. In this subsection, our goal is to extract a subset of content elements $A \subseteq D$ with $|A| \leq K$ such that the following properties are preserved in the extracted subset: (1) **Coverage**: We want the content elements of A to represent the whole document D well. Thus, it is expected that any content element in A to be more similar to many elements in D. (2) **Diversity**: Since the poster is of very limited size, we want to avoid unnecessary redundancy in content present in the poster. This property ensures that any content element in A is very different from most of the other elements in A. (3) **Multimodal Alignment**: The output poster contains both images and text. Images and text contain complementary information. Hence, it is important to ensure that the images and text present in the poster are aligned with each other. For e.g., the text present in the poster can brief about the image. Thus, any image element in A should be similar to some text elements in A and vice-versa. (4) **Ground Truth Data Adaptability**: All the above properties are different intrinsic properties expected in a summarization. However, ground truth summarization data may have other hidden properties which may not be captured above. So, we also want our algorithm to learn from the set of multimodal ground truth summary data.

Thus, A can be considered as an extractive multimodal summary of the multimodal document D where our goal is to design a summarization model which is trainable and equipped with the inductive bias as mentioned above. Next, we construct the following objective function to achieve this.

$$f(A) = \sum_{u \in [d]} w_u \sqrt{\sum_{x \in A} \sum_{y \in D} x_u y_u - \sum_{x \in A} \sum_{y \in A} x_u y_u + \sum_{x \in A_I} \sum_{y \in A_T} x_u y_u + |D| \sum_{x \in A} x_u} \quad (1)$$

Here, $[d] = \{1, 2, \cdots, d\}$ denotes the set of dimensions of $\mathbb{R}^d$. We use a vector of trainable weight parameters $w = [w_1, w_2, \cdots, w_d]^T \geq 0$ which intuitively captures the importance of each dimension of the embedding space. Ideally for a given document D, we would like to choose the subset $A \subseteq D$ which maximizes $f(A)$. The term $\sum_{x \in A} \sum_{y \in D} x_u y_u$ captures the similarity of an element $x \in A$ to an element $y \in D$ for the uth dimension. Since both x and y are L1-normalized, this term over the outer summation contributes more when x and y are similar to each other. Thus, the first term captures the notion of coverage. Similarly, the second term $\sum_{x \in A} \sum_{y \in A} x_u y_u$ captures the similarity of content elements within the extracted multimodal summary A. Thus, the negation of this term can be considered as the diversity of the elements within A. So far, we have not differentiated between the text and images within A. However, selected images in A need to be aligned with the text present in the poster. The third term $\sum_{x \in A_I} \sum_{y \in A_T} x_u y_u$ in Equation 1 ensures multimodal alignment, where A_I is the set of images in A and A_T is the set of text sentences in A. The last term $|D| \sum_{x \in A} x_u$ is introduced to ensure some nice mathematical property of our loss function. Since, $x \in D$ are L1-normalized, the last term is a constant if $w_u = \frac{1}{d}, \forall u \in [d]$ (initial condition as discussed in Section 3.2).

Next, we aim to to design a loss function to train the parameters w of our model. As mentioned in Section 3, we are given with a training set of multimodal documents with the corresponding ground truth summaries as $\mathcal{T} = \{(D_1, A_1^*), (D_1, A_1^*), \cdots, (D_M, A_M^*)\}$. For any $(D_i, A_i^*) \in \mathcal{T}$, we want the value of the function f on our model generated summary to be close to $f(A_i^*)$. We consider the following hinge loss function here:

$$\min_{w \geq 0} \sum_{i=1}^{M} \left(\max \left(f(A_i^*) - \max_{\substack{A \subseteq D_i \\ |A| \leq K}} \{f(A)\}, 0 \right) + \frac{\lambda}{2} ||w||_2^2 \right) \quad (2)$$

In the above equation, we also use an L2 regularizer on w with a weight hyperparameter $\lambda \geq 0$. Based on the performance on validation set, we keep $\lambda = 0.1$ for all our experiments. The predicted summary is obtained by maximizing $f(A)$ w.r.t. A such that $A \subseteq D$ with $|A| \leq K$. Minimizing the hinge loss w.r.t. the trainable non-negative weight parameters w ensures that the maximum of $\{f(A)\}$ (i.e., on model

predicted summary) is not too far less from the ground truth summary on the training data. We discuss the solution strategy and training of this optimization problem next.

3.2 Training and Optimization

The optimization function in Equation 2 is a constrained min-max optimization on two different types of variables. Here, w is continuous and non-negative. But A is a subset of with a fixed cardinality. To solve this, we use an iterative alternating optimization strategy as discussed below.

Maximization w.r.t. A Let us first focus on maximizing the objective w.r.t. A while keeping w fixed. Then it is essentially a subset selection problem for each D_i, $\forall i = 1, 2, \cdots, M$. Subset selection problems are typically combinatorial in nature and often computationally infeasible. But the following theorem shows an important property of f which will help us to solve the optimization problem.

Theorem 3.1. ***The set function f in Equation 1 is a monotone submodular function.***

The proof is included in the supplementary [2]. In contrast to our proposed function in Equation 1, most of the existing submodular functions (Lin and Bilmes 2011; Modani et al. 2016) for text summarization capture diversity rewards assuming the availability of cluster of text, ignore the multimodal aspect of the problem and are not trainable. To maximize $f(A)$, we use the simple greedy algorithm which is a $(1 - \frac{1}{e})$ factor approximation of the optimal solution since f is monotone and submodular (Lin and Bilmes 2011). At each step of the greedy algorithm, we include a new content element $x \in D \setminus A$ into A for which $f(A \cup \{x\}) - f(A)$ is maximum till $|A| = K$.

Minimization w.r.t. w Next, we assume A to be fixed and use a projected stochastic gradient descent approach to minimize the loss function in Equation 2 w.r.t $w \geq 0$. The subgradient of the loss w.r.t. w_u (u^{th} dimension) on the i-th sample of the training set is calculated as $\frac{\partial f(A_i^*)}{\partial w_u} - \frac{\partial f(A)}{\partial w_u} + \lambda w_u$, where $A = \max_{\substack{A \subseteq D_i \\ |A| \leq K}} \{f(A)\}$. This gives us the stochastic gradient descent step with a learning rate $\alpha > 0$ as (check proof of Theorem 3.1):

$$w_u = w_u - \alpha\Big(\sqrt{\Big(\sum_{x \in A_i^*} x_u\Big)\Big(|D| + \sum_{y \in D, y \notin A_i^*} y_u\Big) + h(A_i^*)} - \sqrt{\Big(\sum_{x \in A} x_u\Big)\Big(|D| + \sum_{y \in D, y \notin A} y_u\Big) + h(A) + \lambda w_u}\Big)$$

To ensure non-negativity, we project it to the positive quadrant by setting $w_u = \max(0, w_u), \forall u = 1, 2, \cdots, d$.

We iteratively solve the optimization problem in Equation 2 by maximizing it w.r.t. A by keeping w fixed, and then minimizing it w.r.t. w by keeping A fixed. We repeat these two steps until the loss converges on the validation set used in the experiments.

To calculate the value of $f(A)$, we use a weighted sum of d terms and each term can be calculated using precomputing and using the corresponding previous term. Since we are using the greedy algorithm to find $max_{A \subseteq D_i} f(A)$, this total step has time complexity of $O(KNd)$. For the update of the weights step, we do d updates where all weights get updated and for each update, all the summations can be done independently and then multiplied so we get the time complexity for each update as $O(K)$ ($|D|$ and total sum ($\Sigma_{y \in D} y_u$) values are pre-computed hence the only term calculated during run time is $\Sigma_{x \in A} x_u$ term, which is then used to calculate $\Sigma_{y \in D, y \notin A} y_u = \Sigma_{y \in D} y_u - \Sigma_{x \in A} x_u$), this gives us total training time complexity as $O(NKd + Kd)$ for one full training update. For the inference time, we only need to use greedy algorithm to get the best possible subset (summary), hence this is will be of time complexity $O(KNd)$

3.3 Content Paraphrasing

Text sentences in the multimodal extractive summary may not be suitable to put in the poster directly. We choose ChatGPT (GPT-3.5-turbo) (OpenAI 2022) as it is shown to perform very well for text paraphrasing for different use cases (Chen et al. 2023) and less expensive compared to GPT-4 (OpenAI 2023). But, applying it directly to the long text of the whole document is not always possible due to the length of the document. However, the length of the extractive summary is limited to K. We choose K in such a way that the whole text from the extracted multimodal summary can be fed to ChatGPT within a single API call. Please refer to our supplementary material for sample prompts used for rephrasing content.

We use the output paraphrased text along with the images selected in the multimodal extracted summary as the content to be put in the poster.

3.4 Template Generation

We define the template to be a combination of different style elements such as font of the text and different colors to be used, and the layout of the poster (position of different content elements). We discuss each of them in more detail in subsequent sections.

Font Selection Fonts are crucial components of posters as they signal the intent of the content provided and enable mental maps for familiarity. For example, cooking books are often associated with serif and cursive styles. In this regard, we train a model based on the poster title guiding the visual attributes of the font chosen. The dataset for this task was the *Let me choose* dataset (Shirani et al. 2020), which consists of pairs of titles and the most appropriate fonts for them, from a sample size of 10 fonts. We use a fine-tuned MiniLM (Wang et al. 2020) transformer model as the base encoder to the font selection model. The 384-dimensional feature vector is passed through a 2-layer fully connected network with a dropout layer and uses the LeakyReLU activation function. To find the appropriate learning rate for this process, we use LR-Finder [3] and trained the model for 20,000 epochs on the *train* section of the dataset.

[2] The full version is present at https://arxiv.org/abs/2405.20213

[3] https://github.com/davidtvs/pytorch-lr-finder

Figure 2: Screenshot of a poster generated by PostDoc

Color Selection Colors are also important for posters as they capture attention and contextualise content. Our pipeline has three main colors: (1) Background color of the poster; (2) Box fill color that serves as the color for the bounding boxes that house textual content; and (3) Text fill color - the font color of the texts. To generate the colors used in the poster, we use a model trained on the TPN architecture as proposed in Text2Colors (Bahng et al. 2018). This model, called *TPNSmall*, is trained for 7,000 epochs on the *Text2Colors* dataset. For a given poster title, we first pass it through a publicly available QA (Question Answering) model [4] with the prompt *"What is the main point here?"*. This gives the *intent* of the poster. TPNSmall outputs a palette of hex codes, given the intent.

The dominant color in this palette is chosen to be the background color of the poster, and its complement is chosen to be the box fill color. The text text fill color is chosen to be black or white based on its contrast with the box fill color. We then use the background color in a prompt to Firefly [5] that generates a background grounded on it.

Layout Generation We propose a heuristic based approach for generating a balanced layout conditioned on the paraphrased content. For each topic with the associated bullet points from the paraphrased content, we create a text box in the poster layout. Similarly, for each image (with the associated caption when available), we create an image box. We kept the images on the left and text boxes on the right by dividing the space into two parts vertically. The width of the text boxes is fixed where as that of the images is adjusted based on number of images. To estimate the height of the text boxes, we consider the content length. Detailed calculations for the same are provided in the supplementary material. By following this approach, we achieve a well-organized and visually appealing layout for posters, adapting the design based on the number of text and image boxes required. A sample poster is shown in Figure 2.

[4]https://huggingface.co/deepset/roberta-base-squad2

[5]https://firefly.adobe.com/

4 Experimental Analysis

In this section, we discuss the details about the experimental setup and results obtained from both automated and human evaluation to understand the quality of the posters generated from PostDoc.

4.1 Datasets Used

We use the MSMO Dataset collected by Zhu et al. (2018) for training and testing our method for multimodal summarization. The MSMO Dataset has 312,581 samples of which only the test set (10,261 samples) has image annotations present as part of the multimodal summary. We require image annotations for training our deep submodular function. So, we use 9000 samples from the test set for our training, 261 samples for validation and the remaining 1000 samples for testing.

In order to test the content generated by our multimodal summarization module to that with the actual posters, we make use of the NJU-Fudan Dataset (Qiang et al. 2019). It contains 85 pairs of scientific papers and their corresponding posters. We extract the text and images from the papers and posters using Adobe Extract. We filter paper-poster pairs so that they each have at least one image after being processed by Extract. After this step, we have a filtered dataset of 76 paper-poster pairs. We do not use any portion of this dataset for training. We use it only for testing our summarization method on out of domain data to analyze the generalization ability. We report the performance on MSMO and NJU-Fudan datasets in Table 1.

For training the font selection model, we make use of the Let Me Choose Dataset (Shirani et al. 2020) as discussed in Section 3.4. It contains 1,309 short texts which are mapped to one of 10 fonts. We follow the same split mentioned by the authors for training (70%),validation (10%) and testing (20%).

4.2 Baseline Methods

We could not find any replicable source code for existing document-to-poster works (Qiang et al. 2019; Xu and Wan 2021, 2022) to make an end-to-end comparison. So, for a thorough evaluation, we analyze each module of PostDoc with the corresponding baselines.

Multimodal Summarization To the best of our understanding, there are no publicly available replicable implementation for any of the existing *multimodal-in and multimodal-out* summarization approaches (Zhu et al. 2018, 2020; Zhang et al. 2021, 2022). Additionally, as we are using a subset of the MSMO test set (§4.1) for training, we will not be able to carry over the metrics reported in these works. So, we compare our performance with baselines for text summarization and include images in the summary as a post-processing step. For the text summarization, we use the following.

Method	MSMO Datset ROUGE-L	Coverage	Diversity	IP	Inference Time	NJU-Fudan Dataset ROUGE-L	Coverage	Diversity	IP	Inference Time
Memsum + BLIP	0.24 ± 0.11	0.29 ± 0.06	0.31 ± 0.11	0.73 ± 0.33	3.27	0.27 ± 0.03	0.38 ± 0.05	0.64 ± 0.05	0.39 ± 0.28	10.35
BRIO + BLIP	0.36 ± 0.11	0.37 ± 0.06	0.45 ± 0.20	0.75 ± 0.32	1.04	0.07 ± 0.02	0.27 ± 0.06	$\mathbf{0.66 \pm 0.06}$	0.38 ± 0.26	6.09
GPT-3.5 + BLIP	0.27 ± 0.08	$\mathbf{0.38 \pm 0.06}$	0.54 ± 0.19	$\mathbf{0.75 \pm 0.32}$	14.88	0.13 ± 0.05	0.31 ± 0.06	0.65 ± 0.05	0.39 ± 0.28	47.37
PostDoc	$\mathbf{0.68 \pm 0.14}$	0.30 ± 0.03	$\mathbf{0.58 \pm 0.06}$	0.74 ± 0.34	**0.68**	$\mathbf{0.50 \pm 0.04}$	$\mathbf{0.42 \pm 0.03}$	0.61 ± 0.04	$\mathbf{0.53 \pm 0.32}$	**4.25**

Table 1: Comparison of various multimodal summarization methods on the MSMO (Zhu et al. 2018) and NJU-Fudan (Qiang et al. 2019) datasets

Method	MSMO Datset ROUGE-L	Coverage	Diversity	IP	NJU-Fudan Dataset ROUGE-L	Coverage	Diversity	IP
PostDoc w/o dsf	0.57 ± 0.13	0.23 ± 0.04	0.61 ± 0.09	0.71 ± 0.44	0.43 ± 0.10	0.34 ± 0.07	0.65 ± 0.07	0.37 ± 0.17
PostDoc w/o coverage	0.51 ± 0.08	0.24 ± 0.06	0.61 ± 0.14	$\mathbf{0.75 \pm 0.27}$	0.48 ± 0.05	$\mathbf{0.43 \pm 0.04}$	0.57 ± 0.06	0.52 ± 0.32
PostDoc w/o diversity	0.50 ± 0.13	0.24 ± 0.03	0.62 ± 0.14	0.75 ± 0.27	0.48 ± 0.05	0.43 ± 0.04	0.58 ± 0.06	0.53 ± 0.32
PostDoc w/o alignment	0.56 ± 0.12	0.25 ± 0.06	$\mathbf{0.63 \pm 0.13}$	0.75 ± 0.28	0.45 ± 0.04	0.36 ± 0.03	$\mathbf{0.67 \pm 0.03}$	0.34 ± 0.19
PostDoc	$\mathbf{0.68 \pm 0.14}$	$\mathbf{0.30 \pm 0.03}$	0.58 ± 0.05	0.74 ± 0.34	$\mathbf{0.50 \pm 0.04}$	0.42 ± 0.03	0.61 ± 0.04	$\mathbf{0.53 \pm 0.32}$

Table 2: Model ablation study of PostDoc on MSMO and NJU-Fudan Datasets

Font Recommendation Method	Top-1 F1	Top-3 F1	Layout Generation Method	NGOMetric
BERT Model	0.2697	0.5191	LayoutDM	0.27
PostDoc	**0.4301**	**0.5950**	PostDoc	**0.46**

Table 3: Results of font recommendation on Let Me Choose Dataset and conditional layout generation on the NJU-Fudan Dataset

1. **Extractive Summarization**: We use the publicly available **MemSum** architecture (Gu, Ash, and Hahnloser 2021) which currently has the SOTA performance on the GovReport dataset (Huang et al. 2021).
2. **Abstractive Summarization**: We use the publicly available **BRIO** (Liu et al. 2022) architecture, which currently has the SOTA performance on the CNN/Daily Mail dataset (Nallapati et al. 2016)
3. **GPT-3.5-turbo**: In this baseline, we chunk the input text and summarize each chunk with GPT-3.5-turbo. We concatenate the summaries and finally paraphrase it further with another GPT-3.5-turbo call.

With each of the above text summaries, the images are included in the summary by choosing the top K_I images from the input document based on their similarity with the summarized text. Here the similarity is calculated by the cosine of the BLIP embeddings of text and images. From the training set, we calculate the average ratio of the number of images in the summary and that in the input document. To fix K_I during inference on a document, we multiply the number of images present in the document with that ratio.

Template Generation For font selection, we compare our method (§3.4) with the Let Me Choose BERT Model (Shirani et al. 2020). For layout generation, we compare our method (§3.4) with LayoutDM (Inoue et al. 2023) on the test section of the NJU-Fudan Dataset.

4.3 Evaluation Metrics

We evaluate the generated posters on the following aspects.

Multimodal Summarization To evaluate the salience of the text generated by our method, we employ the standard text summarization metric ROUGE-1, ROUGE-2 and ROUGE-L which compares the generated summary with the ground truth. To evaluate the generated multimodal summary, we use coverage and diversity (Kothawade et al. 2020) along with image precision and image recall. Coverage is measured between the generated multimodal summary and the multimodal source document. We define $Coverage(A) = \frac{1}{|D||A|}\sum_{x\in D}\sum_{y\in A} cosine(x,y)$ and $Diversity(A) = 1 - \frac{1}{|A|^2}\sum_{x,y\in A} cosine(x,y)$. Here, D contains the BLIP embeddings for all the content elements of the input multimodal document and A contains the BLIP embeddings from the generated summary. An ideal summary will have high coverage and high diversity. Following Zhu et al. (2018), we use image precision I_P to evaluate the images selected by our method. $I_P = \frac{|I_A \cap I_G|}{|I_A|}$, where I_A and I_G refer to the images in generated summary and the images in the ground truth summary. We also report average inference time as a metric for computational overhead.

Font Selection Following Shirani et al. (2020), we measure the performance of font selection models using the average weighted F1-Score over top k where $k = \{1, 3\}$.

Layout Generation To evaluate the layouts generated, we use a combination of the following NGOMetrics [6]: We use a weighted combination of the equilibrium of the bounding boxes, padding in the layout, density of the bounding boxes with respect to the overall layout, and overlap between the bounding boxes. This allows us to score layouts based on their aesthetic properties. Please refer to the details about the computation of these metrics in the supplementary material.

4.4 Performance Analysis

Multimodal Summarization Table 1 shows the mean and standard deviation of the performance of PostDoc and the

[6] http://www.mi.sanu.ac.rs/vismath/ngo/index.html

baselines on MSMO and NJU-Fudan datasets. By investigating the results, we note the following: **(1)**: PostDoc outperforms all of the baselines on the basis of the ROUGE metrics with significant margins. This shows that the our multimodal summarization module captures relevant information from the source document which aligns with the ground truth. **(2)**: Regarding the visual modality metric image precision, our model performs on par or slightly worse than the baselines on MSMO dataset, but significantly outperforms all the baselines on NJU-Fudan dataset. It is important to note that the number of images selected in the baseline is pre-fixed based on the average number of images present in the document on the training set. However, for PostDoc, we do not differentiate between the text and images in the selection criteria during the inference. This gives a benefit to the baselines on the MSMO dataset for image selection. **(3)**: As mentioned in Section 4.2, we consider extractive summarization (MemSum), abstractive summzarization (BRIO) and GPT-3.5-turbo for the text summarization module of the baselines. MemSum and BRIO are trained on 17,517 and 200,000 samples respectively and GPT-3.5-turbo is trained on large text databases available on the internet. Our model is able to perform competitively with these baselines even though the data it is trained on (9,000 samples) is significantly lesser than the training data of these text summarization methods. **(4)**: The GPT-3.5+BLIP baseline performs competitively on coverage and diversity. But the latency and the cost associated with GPT calls is very high. On average, the number of tokens processed by GPT-3.5 for MSMO dataset is 838.19 and the NJU-Fudan dataset is 5323.85. PostDoc is much faster (more than **20x** and **10x** compared to GPT-3.5+BLIP on MSMO and NJU-Fudan datasets respectively) than all the baselines for average inference time per document.

Template Generation Table 3 shows the performance of our model and the baseline on the Let Me Choose dataset for font selection. Our model outperforms the baseline on both the Top-1 F1 Score and the Top-3 F1 Score. It also compares our layout generation module with LayoutDM. As LayoutDM wasn't explicitly trained on posters, we initially generate 250 candidate layouts using LayoutDM and choose the layout which gives the highest equi-weighted NGOMetrics Score. Our layout generation module, which relies on heuristics, achieves a score that is almost twice the score achieved by LayoutDM; please refer to our supplementary material for individual metrics' scores.

4.5 Model Ablation Study

To understand the importance of different components of PostDoc, we perform the following ablation experiments on MSMO and NJU-Fudan datasets, and report the results in Table 2.

PostDoc w/o DSF: Instead of using the deep submodular function proposed in Equation 1, we make use of a simple feature-based submodular function (without any trainable parameters) which takes coverage and diversity into account: $f(A) = \lambda \sum_{x \in A} \sum_{y \in D} x_u y_u - \sum_{x \in A} \sum_{y \in A} x_u y_u$. For inference we choose the subset A which gives the maximum value of $f(A)$. The value of λ was set as 0.2 for this experiment.

We remove the terms $\sum_{x \in A} \sum_{y \in D} x_u y_u$, $\sum_{x \in A} \sum_{y \in A} x_u y_u$, and $\sum_{x \in A_I} \sum_{y \in A_T} x_u y_u$ respectively from Equation 1 and proceed with the min-max optimization procedure (Section 3.2) to formulate the ablations *PostDoc w/o Coverage*, *PostDoc w/o Diversity*, and *PostDoc w/o Alignment*.

From Table 2, we can see that PostDoc is able to achieve the best performance on ROUGE metrics which shows the importance of the combination of all the components present in it. On the other three metrics, we do not see any consistent pattern among the different model variants.

Questions	GPT-3.5 + BLIP	PostDoc
Coverage	3.13 ± 0.83	$\mathbf{3.40 \pm 0.64}$
Duplication	$\mathbf{4.26 \pm 0.27}$	4.13 ± 0.37
Content Ordering	3.33 ± 0.62	3.33 ± 0.47
Image Selection	2.66 ± 0.47	$\mathbf{3.0 \pm 0.70}$
Template	2.33 ± 0.5	$\mathbf{3.46 \pm 0.18}$
Run Time	1.86 ± 0.29	$\mathbf{3.60 \pm 0.64}$

Table 4: Results of human eval on a subset of NJU-Fudan

4.6 Human Evaluation

We have conducted a small-scale human survey to understand the quality of the generated posters from the actual human perspective. We hired 3 experts in AI as reviewers for this task. We randomly chose 5 research papers from NJU-Fudan dataset. We used GPT-3.5+BLIP as the only baseline for this study as GPT-3.5 is often considered to be close to humans in generative tasks. Each reviewer generated the posters by the two algorithms from each of the selected 5 documents and gave a rating on a scale of 1 (worst) to 5 (best) to each poster for each of the following aspects: (1) *Coverage* of the document? (2) *Duplication* of content? (3) *Ordering of content*? (4) *Selected images*? (5) *Template* of the poster? (6) *Run time* of the algorithm? In Table 4, we compute average and standard deviation of the ratings provided by all the reviewers on all the documents. The human evaluation shows that PostDoc is as per or better in terms of output content quality and much better in terms of user satisfaction with the layout and runtime.

5 Conclusion

We have presented PostDoc, an end-end pipeline to automatically generate a poster from a long multimodal document. It is interesting to find that PostDoc, by using a novel combination of deep submodular functions and a single call to LLM (ChatGPT) is able to achieve better or comparable performance than direct calls to the same LLM which is expensive both in terms of computation and cost. In the current work, the performance of PostDoc is limited for non-natural images such as flow-chart and neural diagrams, and other structured elements like tables, etc. We plan to fine-tune VLMs on documents containing such elements as a future work.

References

Bahng, H.; Yoo, S.; Cho, W.; Park, D. K.; Wu, Z.; Ma, X.; and Choo, J. 2018. Coloring with Words: Guiding Image Colorization Through Text-Based Palette Generation. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XII*, 443–459. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-030-01257-1.

Bilmes, J.; and Bai, W. 2017. Deep Submodular Functions. arXiv:1701.08939.

Chai, S.; Zhuang, L.; and Yan, F. 2023. LayoutDM: Transformer-based Diffusion Model for Layout Generation. arXiv:2305.02567.

Chen, X.; Ye, J.; Zu, C.; Xu, N.; Zheng, R.; Peng, M.; Zhou, J.; Gui, T.; Zhang, Q.; and Huang, X. 2023. How Robust is GPT-3.5 to Predecessors? A Comprehensive Study on Language Understanding Tasks. *arXiv preprint arXiv:2303.00293*.

Fu, T.-J.; Wang, W. Y.; McDuff, D.; and Song, Y. 2022. DOC2PPT: Automatic Presentation Slides Generation from Scientific Documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1): 634–642.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; and Wang, H. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Gu, N.; Ash, E.; and Hahnloser, R. H. 2021. MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes. *arXiv preprint arXiv:2107.08929*.

Han, Z.; Gao, C.; Liu, J.; Zhang, S. Q.; et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

He, B.; Wang, J.; Qiu, J.; Bui, T.; Shrivastava, A.; and Wang, Z. 2023. Align and Attend: Multimodal Summarization With Dual Contrastive Losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14867–14878.

Huang, L.; Cao, S.; Parulian, N.; Ji, H.; and Wang, L. 2021. Efficient attentions for long document summarization. *arXiv preprint arXiv:2104.02112*.

Inoue, N.; Kikuchi, K.; Simo-Serra, E.; Otani, M.; and Yamaguchi, K. 2023. LayoutDM: Discrete Diffusion Model for Controllable Layout Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10167–10176.

Islam, R.; and Moushi, O. M. 2024. GPT-4o: The Cutting-Edge Advancement in Multimodal LLM. *Authorea Preprints*.

Kothawade, S.; Girdhar, J.; Lavania, C.; and Iyer, R. 2020. Deep submodular networks for extractive data summarization. *arXiv preprint arXiv:2010.08593*.

Li, H.; Zhu, J.; Ma, C.; Zhang, J.; and Zong, C. 2017. Multi-modal summarization for asynchronous collection of text, image, audio and video. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1092–1102.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 12888–12900. PMLR.

Lin, H.; and Bilmes, J. 2011. A Class of Submodular Functions for Document Summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 510–520. Portland, Oregon, USA: Association for Computational Linguistics.

Lin, J.; Guo, J.; Sun, S.; Yang, Z.; Lou, J.-G.; and Zhang, D. 2024. Layoutprompter: Awaken the design ability of large language models. *Advances in Neural Information Processing Systems*, 36.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Liu, Y.; Liu, P.; Radev, D.; and Neubig, G. 2022. BRIO: Bringing Order to Abstractive Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2890–2903.

Mademlis, I.; Tefas, A.; Nikolaidis, N.; and Pitas, I. 2016. Multimodal stereoscopic movie summarization conforming to narrative characteristics. *IEEE Transactions on Image Processing*, 25(12): 5828–5840.

Maheshwari, H.; Bandyopadhyay, S.; Garimella, A.; and Natarajan, A. 2024. Presentations are not always linear! GNN meets LLM for Text Document-to-Presentation Transformation with Attribution. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 15948–15962.

Meng, F.; Wang, J.; Li, C.; Lu, Q.; Tian, H.; Liao, J.; Zhu, X.; Dai, J.; Qiao, Y.; Luo, P.; et al. 2024. MMIU: Multimodal Multi-image Understanding for Evaluating Large Vision-Language Models. *arXiv preprint arXiv:2408.02718*.

Modani, N.; Maneriker, P.; Hiranandani, G.; Sinha, A. R.; Utpal; Subramanian, V.; and Gupta, S. 2016. Summarizing multimedia content. In *Web Information Systems Engineering–WISE 2016: 17th International Conference, Shanghai, China, November 8-10, 2016, Proceedings, Part II 17*, 340–348. Springer.

Nallapati, R.; Zhou, B.; dos Santos, C.; Gulcehre, C.; and Xiang, B. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 280–290. Berlin, Germany: Association for Computational Linguistics.

OpenAI. 2022.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Petroni, F.; Lewis, P.; Piktus, A.; Rocktäschel, T.; Wu, Y.; Miller, A. H.; and Riedel, S. 2020. How Context Affects Language Models' Factual Predictions. In *Automated Knowledge Base Construction*.

Qiang, Y.-T.; Fu, Y.-W.; Yu, X.; Guo, Y.-W.; Zhou, Z.-H.; and Sigal, L. 2019. Learning to generate posters of scientific papers by probabilistic graphical models. *Journal of Computer Science and Technology*, 34: 155–169.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Shelledy, D. C. 2004. How to make an effective poster. *Respiratory Care*, 49(10): 1213–1216.

Shirani, A.; Dernoncourt, F.; Echevarria, J.; Asente, P.; Lipka, N.; and Solorio, T. 2020. Let Me Choose: From Verbal Context to Font Selection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Sun, E.; Hou, Y.; Wang, D.; Zhang, Y.; and Wang, N. X. 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*, 1405–1418.

Sun, T.; Shao, Y.; Qian, H.; Huang, X.; and Qiu, X. 2022. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, 20841–20855. PMLR.

Tschiatschek, S.; Iyer, R. K.; Wei, H.; and Bilmes, J. A. 2014. Learning mixtures of submodular functions for image collection summarization. *Advances in neural information processing systems*, 27.

Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. *CoRR*, abs/2002.10957.

Xu, S.; and Wan, X. 2021. Neural content extraction for poster generation of scientific papers. *arXiv preprint arXiv:2112.08550*.

Xu, S.; and Wan, X. 2022. PosterBot: A System for Generating Posters of Scientific Papers with Neural Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11): 13233–13235.

Zhang, J.; Zhao, Y.; Saleh, M.; and Liu, P. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, 11328–11339. PMLR.

Zhang, L.; Zhang, X.; Pan, J.; and Huang, F. 2021. Hierarchical Cross-Modality Semantic Correlation Learning Model for Multimodal Summarization. arXiv:2112.12072.

Zhang, Z.; Meng, X.; Wang, Y.; Jiang, X.; Liu, Q.; and Yang, Z. 2022. UniMS: A Unified Framework for Multimodal Summarization with Knowledge Distillation. arXiv:2109.05812.

Zhong, M.; Liu, P.; Chen, Y.; Wang, D.; Qiu, X.; and Huang, X.-J. 2020. Extractive Summarization as Text Matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6197–6208.

Zhu, J.; Li, H.; Liu, T.; Zhou, Y.; Zhang, J.; and Zong, C. 2018. MSMO: Multimodal Summarization with Multimodal Output. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4154–4164. Brussels, Belgium: Association for Computational Linguistics.

Zhu, J.; Zhou, Y.; Zhang, J.; Li, H.; Zong, C.; and Li, C. 2020. Multimodal Summarization with Guidance of Multimodal Reference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05): 9749–9756.