DOC2PPT: Automatic Presentation Slides Generation from Scientific Documents

Tsu-Jui Fu, William Yang Wang
UC Santa Barbara
{tsu-juifu,william}@cs.ucsb.edu

Daniel McDuff, Yale Song
Microsoft Research
{damcduff,yalesong}@microsoft.com

Abstract

Creating presentation materials requires complex multimodal reasoning skills to summarize key concepts and arrange them in a logical and visually pleasing manner. Can machines learn to emulate this laborious process? We present a novel task and approach for document-to-slide generation. Solving this involves document summarization, image and text retrieval, slide structure and layout prediction to arrange key elements in a form suitable for presentation. We propose a hierarchical sequence-to-sequence approach to tackle our task in an end-to-end manner. Our approach exploits the inherent structures within documents and slides and incorporates paraphrasing and layout prediction modules to generate slides. To help accelerate research in this domain, we release a dataset about 6K paired documents and slide decks used in our experiments. We show that our approach outperforms strong baselines and produces slides with rich content and aligned imagery.

1. Introduction

Creating presentations is often a work of art. It requires skills to abstract complex concepts and conveys them in a concise and visually pleasing manner. Consider the steps involved in creating presentation slides based on a white paper or manuscript: One needs to 1) establish a storyline that will connect with the audience, 2) identify essential sections and components that support the main message, 3) delineate the structure of that content, e.g., the ordering/length of the sections, 4) summarize the content in a concise form, e.g., punchy bullet points, 5) gather figures that help communicate the message accurately and engagingly, and 6) arrange these elements (e.g., text, figures, and graphs) in a logical and aesthetically pleasing manner on each slide.

Can machines emulate this laborious process by *learning* from the plethora of example manuscripts and slide decks created by human experts? We argue that this is an area where AI can enhance humans' productivity, e.g., by drafting slides for humans to build upon. This would open up new opportunities to human-AI collaboration, e.g., one could quickly generate a slide deck by revising the draft or simply generate slide decks of many papers and skim them through to digest a lot of material quickly.

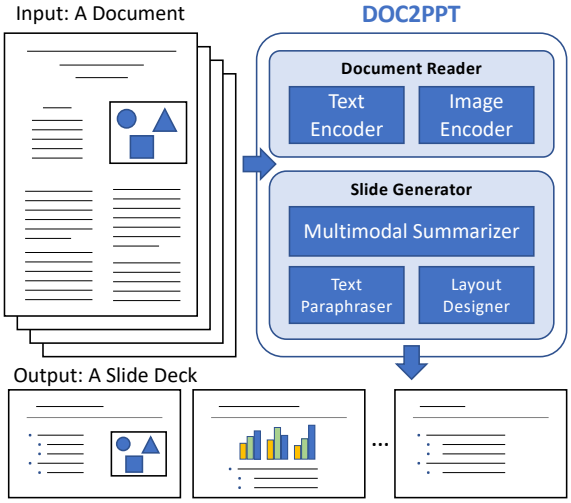


Figure 1. We introduce a novel task of generating a slide deck from a document. This requires solving several challenges in the vision & language domain, e.g., visual-semantic embedding and multimodal summarization. In addition, slides exhibit unique properties such as concise text (bullet points) and stylized layout. We propose an approach to solving DOC2PPT, tackling these challenges.

However, building such a system poses unique challenges in vision and language understanding. Both the input (a manuscript) and output (a slide deck) contain tightly coupled visual and textual elements; thus, it requires multimodal reasoning. Further, there are significant differences in the presentation: compared to manuscripts, slides tend to be more *concise* (e.g., containing bullet points rather than full sentences), *structured* (e.g., each slide has a fixed screen real estate and delivers one or few messages), and *visual-centric* (e.g., figures are first-class citizens, the visual layout plays an important role, etc.).

Existing literature only partially addresses some of the

Project webpage will be released.

challenges above. Document summarization [13, 16] aims to find a concise text summary of the input, but it does not deal with images/figures and lacks multimodal understanding. Cross-modal retrieval [26, 38] focuses on finding a multimodal embedding space but does not produce summarized outputs. Multimodal summarization [68] deals with both (summarizing documents with text and figures), but it lacks the ability to produce structured output (as in slides). Furthermore, none of the above addresses the challenge of finding an optimal visual layout of each slide. While assessing visual aesthetics have been investigated [49], existing work focuses on photographic metrics for images that would not translate to slides. These aspects make ours a unique task in the vision-and-language literature.

In this paper, we introduce DOC2PPT, a novel task of creating presentation slides from documents. As this is a new task with no existing benchmark, we release code, and a new dataset of 5,873 paired scientific documents and associated presentation slide decks (for a total of about 70K pages and 100K slides, respectively). We present a series of automatic data processing steps to extract useful learning signals from documents and slides. We also introduce new quantitative metrics designed to measure the quality of the generated slides.

To tackle this task, we present a hierarchical recurrent sequence-to-sequence architecture that "reads" the input document and "summarizes" it into a *structured* slide deck. We exploit the inherent structure within documents and slides by performing inference at the section-level (for documents) and at the slide-level (for slides). To make our model end-to-end trainable, we explicitly encode section/slide embeddings and use them to learn a policy that determines *when to proceed* to the next section/slide. Further, we learn the policy in a hierarchical manner so that the network decides which actions to take by considering the structural context, e.g., a decision to create a new slide will depend on the section the model is currently summarizing and the previous slides that it has generated thus far.

To account for the concise nature of text in slides (e.g., bullet points), we incorporate a paraphrasing module that converts document-style full sentences to slide-style phrases/clauses. We show that this module drastically improves the quality of the generated textual content for the slides. In addition, we introduce a text-image matching objective that encourages related text-image pairs to appear on the same slide. We demonstrate that this objective substantially improves figure placement in slides. Lastly, we explore both template-based and learning-based solutions for slide layout design and compare them both quantitatively and qualitatively.

To summarize, our main contributions include: 1) Introducing a novel task, dataset, and evaluation metrics for automatic slide generation; 2) Proposing a hierarchical sequence-to-sequence approach that summarizes a document in a structure output format suitable for slide presentation; 3) Evaluating our approach both quantitatively, using our proposed metrics, and qualitatively based on human evaluation. The task of generating presentation slides presents numerous challenges. We hope that our work will enable researchers to advance the state-of-the-art in the vision-and-language domain.

2. Related Work

Vision and Language. Joint modeling of vision-and-language has been studied from different angles. Image/video captioning [59, 66, 40, 62], visual question answering [36, 5, 6], visually-grounded dialogue generation [20] and visual navigation [60] are all tasks that involve learning relationships between visual imagery and text. Despite this large body of work, there remain many vision and language tasks that have not been addressed, e.g., multimodal document generation such as ours. As argued above, our task brings a new suite of challenges to vision-and-language understanding.

Document Summarization. This task has been tackled from two angles: abstractive [16, 55, 14, 45, 22, 67, 9, 54, 43, 52] and extractive [8, 50, 44, 12, 65, 13, 64]. Our DOC2PPT task involves both abstractive and extractive summarization because it requires a model to extract the key content from a document *and* paraphrase it into a concise form. A task closely related to ours is scientific document summarization [23, 47, 34, 51], but to date that work has only focused on producing text summaries, while we focus on generating multimedia slides. Furthermore, existing datasets in this domain (such as TalkSumm [39] and ScisummNet [63]) are rather small with only about 1K documents each. We propose a large dataset of 5,873 pairs of high-quality scientific documents and slide decks.

Visual-Semantic Embedding Our task involves generating slides with relevant text and figures. Learning text-image similarity has been studied in the visual-semantic embedding (VSE) literature [38, 37, 58, 25, 32, 27, 56]. However, unlike the VSE setting where text instances are known in advance, ours requires simultaneously *generating* text and retrieving the related images at the same time.

Multimodal Summarization This task aims to summarize a document with text and figures into a summary that also contains text and figures. MultiModal Summarization with Multimodal Output (MSMO) [68, 69] applies an attention mechanism to generate a textual summary with related images for news articles. Similarly, our task involves summarizing multimodal documents, but it also involves putting the summary in a structured format such as slides.

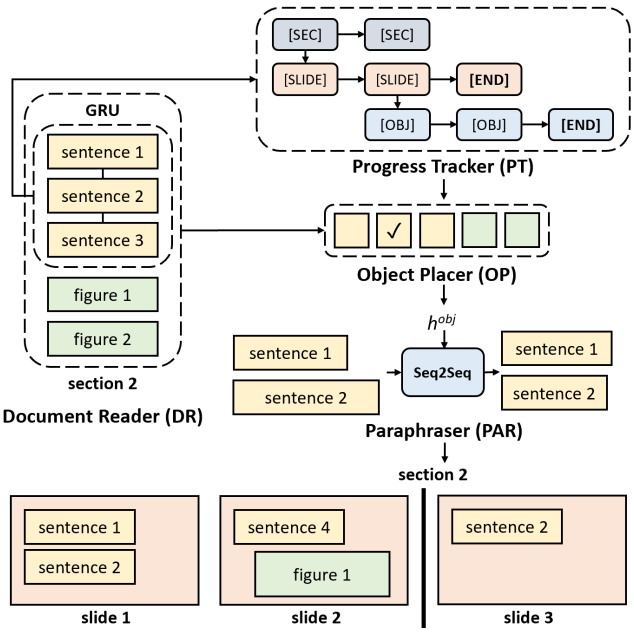


Figure 2. An overview of our network architecture. It consists of four main modules (DR, PT, OP, PAR) that read a document and generate a slide deck in a hierarchically structured manner.

3. Approach

The goal of DOC2PPT is to generate a slide deck from a multimodal document with text and figures.[1] As shown in Fig. 1, the task involves "reading" a document (i.e., encoding sentences and images) and summarizing it, paraphrasing the summarized sentences into a concise format suitable for slide presentation, and placing the chosen text and figures to appropriate locations in the output slides.

Overview Given the multi-objective nature of the task, we design our network with modularized components that are jointly trained in an end-to-end fashion. Fig. 2 shows an overview of our network that includes these modules:

- A **Document Reader (DR)** encodes sentences and figures in a document.
- A **Progress Tracker (PT)** maintains pointers to the input (i.e., which section is currently being processed) and the output (i.e., which slide is currently being generated) and determines when to proceed to the next section/slide based on the progress so far.
- An **Object Placer (OP)** decides which object from the current section (sentence or figure) to put on the current slide. It also predicts the location and the size of each object to be placed on the slide.
- A **Paraphraser (PAR)** takes the selected sentence and rewrites it in a concise form before putting it on a slide.

[1] In this work, figures include images, graphs, charts, and tables.

Notation A document $\mathcal{D}$ is organized into sections $\mathcal{S} = \{S_i\}_{i \in N_S^{in}}$ and figures $\mathcal{F} = \{F_q^{in}\}_{q \in M_F^{in}}$. Each section S_i contains sentences $\mathcal{T}_i^{in} = \{T_{i,k}^{in}\}_{k \in N_i^{in}}$, and each figure $F_q = \{I_q, C_q\}$ contains an image I_q and a caption C_q. We do not assign figures to any particular section because multiple sections can reference the same figure. A slide deck $\mathcal{O} = \{O_j\}_{j \in N_O^{out}}$ contains a number of slides, each containing sentences $\mathcal{T}_j^{out} = \{T_{j,k}^{out}\}_{k \in N_j^{out}}$ and figures $\mathcal{F}_j^{out} = \{F_{j,k}^{out}\}_{k \in M_j^{out}}$. We encode the position and the size of each object on a slide in a bounding box format using an auxiliary layout variable $L_{j,k}$, which includes four real-valued numbers $\{l^x, l^y, l^w, l^h\}$ encoding the x-y offsets (top-left corner), the width and height of a bounding box.

3.1. Model

Document Reader (DR) We extract sentence and figure embeddings from an input document and project them to a shared embedding space so that the Object Placer treats both textual and visual elements as an object coming from a joint multimodal distribution.

For each section S_i, we use RoBERTa [46] to encode each of the sentences $T_{i,k}^{in}$, and then use a bidirectional GRU [17] to extract contextualized sentence embeddings $X_{i,k}^{in}$:

$$\begin{aligned} B_{i,k}^{in} &= \text{RoBERTa}(T_{i,k}^{in}), \\ X_{i,k}^{in} &= \text{Bi-GRU}(B_{i,0}^{in}, ..., B_{i,N_i^{in}-1}^{in})_k, \end{aligned} \tag{1}$$

Similarly, for each figure $F_q^{in} = \{I_q^{in}, C_q^{in}\}$, we apply ResNet-152 [30] to extract the image embedding of I_q^{in} and RoBERTa for the caption embedding of C_q^{in}. We then concatenate them as the figure embedding V_q^{in}:

$$V_q^{in} = [\text{ResNet}(F_q^{in}), \text{RoBERTa}(C_q^{in})]. \tag{2}$$

Next, we project $X_{i,k}^{in}$ and V_q^{in} to a shared embedding space using a two-layer multilayer perceptron (MLP):

$$E_{i,k}^{txt} = \text{MLP}^{txt}(X_{i,k}^{in}), \quad E_q^{fig} = \text{MLP}^{fig}(V_q^{in}). \tag{3}$$

Finally, we combine E_i^{txt} and E^{fig} as the section embedding E_i^{sec} of S_i:

$$E_i^{sec} = \{E_{i,k}^{txt}, E_q^{fig}\}_{k \in N_i^{in}, q \in M_F^{in}} \tag{4}$$

We include all figures $\mathcal{F}$ in *each* section embedding E_i^{sec} because each section can reference any of the figures.

Progress Tracker (PT) We define the PT as a state machine operating in a hierarchically-structured space with sections ([SEC]), slides ([SLIDE]), and objects ([OBJ]). This is to reflect the structure of documents and slides, i.e., each section of a document can have multiple corresponding slides, and each slide can contain multiple objects.

The PT maintains pointers to the current section i and the current slide j, and learns a policy to proceed to the next section/slide as it generates slides. For simplicity, we initialize $i = j = 0$, i.e., the output slides will follow the natural order of sections in an input document.

We construct PT as a three-layer hierarchical RNN with $(\texttt{PT}^{sec}, \texttt{PT}^{slide}, \texttt{PT}^{obj})$, where each RNN encodes the latent space for each level in a section-slide-object hierarchy. This is a natural choice to encode our prior knowledge about the hierarchical structure; in Section 5, we empirically compare this to a "flattened" version of RNN that encodes the section-slide-object structure using a single latent space.

First, $\texttt{PT}^{sec}$ takes as input the head-tail contextualized sentence embeddings from the DR, which encodes the overall information of the current section S_i:

$$h_i^{sec} = \texttt{PT}^{sec}(h_{i-1}^{sec}, [X_{i,1}^{in}, X_{i,N_i^{in}}^{in}]), \tag{5}$$

We use GRU [15] for $\texttt{PT}^{sec}$ and initialize h_0^{sec} to the contextualized sentence embeddings of the first section, i.e., $h_0^{sec} = [X_{0,1}^{in}, X_{0,N_0^{in}-1}^{in}]$.

Based on the section state h_i^{sec}, $\texttt{PT}^{slide}$ models the section-to-slide relationships,

$$a_j^{sec}, h_j^{slide} = \texttt{PT}^{slide}(a_{j-1}^{sec}, h_{j-1}^{slide}, E_i^{sec}), \tag{6}$$

where $h_0^{slide} = h_i^{sec}$, E_i^{sec} is the section embedding (Eq. 4), and a_j^{sec} is a binary action variable that tracks the section pointer, i.e, it decides if the model should generate a new slide for the current section S_i or proceed to the next section S_{i+1}. We implement $\texttt{PT}^{slide}$ as a GRU and a two-layer MLP with a binary decision head that learns a policy ϕ to predict $a_j^{sec} = \{\texttt{[NEW\_SLIDE]}, \texttt{[END\_SEC]}\}$,

$$\begin{aligned} a_j^{sec} &= \text{MLP}_{\phi}^{slide}([h_j^{slide}, \textstyle\sum_r \alpha_{j,r}^{slide} E_{i,r}^{sec}]), \\ \alpha_j^{slide} &= \text{softmax}(h_j^{slide} W (E_i^{sec})^{\intercal}). \end{aligned} \tag{7}$$

Here, $\alpha_j^{slide} \in \mathbb{R}^{N_i^{in}+M^{in}}$ is an attention map over E_i^{sec} that computes the compatibility between h_j^{slide} and E_i^{sec} in a bilinear form.

Finally, the object $\texttt{PT}^{obj}$ tracks which objects to put on the current slide O_j based on the slide state h_j^{slide},

$$\begin{aligned} a_k^{slide}, h_k^{obj} &= \texttt{PT}^{obj}(a_{k-1}^{slide}, h_{k-1}^{obj}, E_i^{sec}), \\ a_k^{slide} &= \text{MLP}_{\psi}^{obj}([h_k^{obj}, \textstyle\sum_r \alpha_{k,r}^{obj} E_{i,r}^{sec}]), \\ \alpha_k^{obj} &= \text{softmax}(h_k^{obj} W (E_i^{sec})^{\intercal}), \end{aligned} \tag{8}$$

where we set $h_0^{obj} = h_j^{slide}$. Similar to $\texttt{PT}^{slide}$, $a_k^{slide} = \{\texttt{[NEW\_OBJ]}, \texttt{[END\_SLIDE]}\}$ is a binary action variable that decides whether to put a new object for the current slide or proceed to the next. We again use a GRU and a two-layer MLP with a policy ψ to implement $\texttt{PT}^{obj}$, together with an attention mechanism that measures the compatibility scores between h_k^{obj} and E_i^{sec}. Note that each of the three PTs have an independent set of weights to ensure that they model distinctive dynamics in the section-slide-object structure.

Object Placer (OP) When $\texttt{PT}^{obj}$ takes an action $a_k^{slide} = \texttt{[NEW\_OBJ]}$, the OP selects an object from the current section S_i and predicts the location on the current slide O_j in which to place it. For this, we use the attention score α_k^{obj} to choose an object (sentence or figure) that has the maximum compatibility score with the current object state h_k^{obj}, i.e., $\arg\max_r \alpha_k^{obj}$. We then employ a two-layer MLP to predict the layout variable for the chosen object,

$$\{l_k^x, l_k^y, l_k^w, l_k^h\} = \text{MLP}^{layout}([h_k^{obj}, \textstyle\sum_r \alpha_{k,r}^{obj} E_{i,r}^{sec}]), \tag{9}$$

Note that the distinctive style of presentation slides requires special treatment of the objects. If an object is a figure, we take only the image part and resize it to fit the bounding box region while maintaining the original aspect ratio. If an object is a sentence, we first paraphrase it into a concise form and also adjust the font size to fit inside the bounding box region.

Paraphraser (PAR) We paraphrase sentences before placing them on slides. This step is crucial because without it the text would be too verbose for a slide presentation.[2] We implement the PAR as an attention-based Seq2Seq [7] with the copy mechanism [28]:

$$\{w_0, ..., w_{l-1}\} = \text{PAR}(T_{j,k}^{out}, h_k^{obj}), \tag{10}$$

where $T_{j,k}^{out}$ is a sentence from a document chosen by OP. We condition PAR on the object state h_k^{obj} to provide contextual information; we provide the importance of this conditioning in the supplementary material.

3.2. Training

We design a learning objective that captures both the structural similarity and the content similarity between the ground-truth slides and the generated slides.

Structural similarity The series of actions a_j^{sec} and a_k^{slide} determines the *structure* of output slides, i.e., the number of slides per section. To encourage our model to generate slide decks with a similar structure as the ground-truth, we define our structural similarity loss as

$$\mathcal{L}_{structure} = \textstyle\sum_j \text{CE}(a_j^{sec}) + \sum_k \text{CE}(a_k^{slide}) \tag{11}$$

where CE is the cross-entropy loss.

[2] In our dataset, sentences in the documents have an average of 17.3 words, while sentences in slides have 11.6 words; the difference is statistically significant ($p = 0.0031$).

	Document & Slide Pairs		Documents			Slides		
	#Pairs	Train / Val / Test	#Sections	#Sentences	#Figures	#Slides	#Sentences	#Figures
CV	2,600	2,073 / 265 / 262	15,588 (6.0)	721,048 (46.3)	24,998 (9.6)	37,969 (14.6)	124,924 (8.0)	4,290 (1.7)
NLP	931	741 / 93 / 97	7,743 (8.3)	234,764 (30.3)	8,114 (8.7)	19,333 (20.8)	63,162 (8.2)	3,956 (4.2)
ML	2,342	1,872 / 234 / 236	17,735 (7.6)	801,754 (45.2)	15,687 (6.7)	41,544 (17.7)	142,698 (8.0)	6,187 (2.6)
Total	5,873	4,686 / 592 / 595	41,066 (6.99)	1,757,566 (42.8)	48,799 (8.3)	98,856 (16.8)	330,784 (8.1)	14,433 (2.5)

Table 1. Descriptive statistics of our dataset. We report both the total count and the average number (in parenthesis).

Content similarity We formulate our content similarity loss to capture various aspects of slide generation quality, measuring whether the model 1) selected important sentences and figures from the input document, 2) adequately phrased sentences in the presentation style (e.g., shorter sentences), 3) placed sentences and figures to the right locations on a slide, and 4) put sentences and figures on a slide that are relevant to each other.

We define our content similarity loss to measure each of the four aspects described above:

$$\mathcal{L}_{content} = \sum\nolimits_k \text{CE}(\alpha_k^{obj}) + \sum\nolimits_l \text{CE}(w_l) + \sum\nolimits_{u,v} \text{CE}(\delta([E_u^{txt}, E_v^{fig}])) + \sum\nolimits_k \text{MSE}(L_k). \quad (12)$$

Selection loss. The first term checks whether the model selected the "correct" objects that also appear in the ground-truth slide deck. This term is slide-insensitive, i.e., the correct/incorrect inclusion of an object is not affected by which specific slide it appears in.

Paraphrasing loss. The second term measures the quality of paraphrased sentences by comparing the output sentence and the ground-truth sentence word-by-word.

Text-image matching loss. The third term measures the relevance of text and figures appearing in the same slide. We follow the literature on visual-semantic embedding [26, 38, 37] and learn an additional multimodal projection head $\delta([E_u^{txt}, E_v^{fig}])$ with a sigmoid activation that outputs a scalar variable in $[0, 1]$ indicating the relevance score of text and figure embeddings. We construct training samples with positive and negative pairs. For positive pairs, we sample text-figure pairs from a) the ground-truth slides and b) paragraph-figure pairs where the figure is mentioned in the paragraph. We randomly construct negative pairs.

Layout loss. The last term measures the quality of slide layout by regressing the predicted bounding box to the ground-truth. While there exist several solutions to bounding box regression [29, 53], we opted for the simple mean squared error (MSE) computed directly over the layout variable $L_k = \{l_k^x, l_k^y, l_k^w, l_k^h\}$.

The final loss We define our final learning objective as

$$\mathcal{L}_{DOC2PPT} = \mathcal{L}_{structure} + \gamma \mathcal{L}_{content} \quad (13)$$

where γ controls the relative importance between structural and content similarity; we set $\gamma = 1$ in our experiments.

To train our model, which is a sequential prediction task, we follow the standard teacher-forcing approach [61] and provide the ground-truth results for the past prediction steps, e.g., the next actions a_j^{sec} and a_k^{slide} are based on the ground-truth actions $\tilde{a}_{j-1}^{sec}$ and $\tilde{a}_{k-1}^{slide}$, the next object α_k^{obj} is selected based on the ground-truth object $\tilde{\alpha}_{k-1}^{obj}$, etc.

3.3. Inference

The inference procedures during training and test times largely follow the same process, with one exception: At test time, we utilize the multimodal projection head $\delta(\cdot)$ to act as a post-processing tool. That is, once our model generates a slide deck, we remove figures that have relevance scores lower than a threshold θ^R and add figures with scores higher than a threshold θ^A. We tune the two hyper-parameters θ^R and θ^A via cross-validation (we set $\theta^R = 0.8$, $\theta^A = 0.9$).

4. Dataset

We collect pairs of documents and the corresponding slide decks from academic proceedings, focusing on three research communities: computer vision (CVPR, ECCV, BMVC), natural language processing (ACL, NAACL, EMNLP), and machine learning (ICML, NeurIPS, ICLR). Table 1 reports the descriptive statistics of our dataset.

Our dataset contains PDF documents and slides in the JPEG image format. For the training and validation set, we automatically extract text and figures from them and perform matching to create document-to-slide correspondences at various levels. To ensure that our test set is clean and reliable, we use Amazon Mechanical Turk (AMT) and have humans perform image extraction and matching for the entire test set. We provide an overview of our extraction and matching processes below; including details of data collection and automatic extraction/matching processes with reliability analyses in the supplementary material.

Text and Figure Extraction. For each document $\mathcal{D}$, we extract sections $\mathcal{S}$ and sentences T^{in} using ScienceParse [4] and figures $\mathcal{F}^{in}$ using PDFFigures2.0 [18]. For each slide deck $\mathcal{O}$, we extract sentences $\mathcal{T}^{out}$ using Azure OCR [1] and figures $\mathcal{F}^{out}$ using morphological transformation and the border following technique [57, 2].

Slide Stemming. Many slides are presented with animations, and this makes $\mathcal{O}$ contain some successive slides that have similar content minus one element on the preceding slide. For simplicity we consider these near-duplicate slides as redundant and remove them by comparing text and image contents of successive slides: if O_{j+1} covers more than 80% of the content of O_j (per text/visual embeddings) we discard it and keep O_{j+1} as it is deemed more complete.

Slide-Section Matching. We match slides in a deck to the sections in the corresponding document so that a slide deck is represented as a set of non-overlapping slide groups each with a matching section in the document. To this end, we use RoBERTa [46] to extract embeddings of the text content in each slide and the paragraphs in each section of the document. We assume that a slide deck follows the section order of the corresponding document, and use dynamic programming to find slide-to-section matching based on the cosine similarity between text embeddings.

Sentence Matching. We match sentences from slides to the corresponding document. We again use RoBERTa to extract embeddings of each sentence in slides and documents, and search for the matching sentence based on the cosine similarity. We limit the search space only within the corresponding sections using the slide-section matching result.

Figure Matching. Lastly, we match figures from slides to those in the corresponding document. We use MobileNet [31] to extract visual embeddings of all I^{in} and I^{out} and match them based on the highest cosine similarity. Note that some figures in slides do not appear in the corresponding document (and hence no match). For simplicity, we discard F^{out} if its highest visual embedding similarity is lower than a threshold $\theta^I = 0.8$.

5. Experiments

`DOC2PPT` is a new task with no established evaluation metrics and baselines. To enable large-scale evaluation we propose automatic metrics specifically designed for evaluating slide generation methods. We carefully ablate various components of our approach and evaluate them on our proposed metrics. We also perform human evaluation to assess the generation quality.

5.1. Evaluation Metrics

Slide-Level ROUGE (ROUGE-SL) To measure the quality of text in the generated slides, we adapt the ROUGE score [41] widely-used in document summarization. Note that ROUGE does not account for the text length in the output, which is problematic for presentation slides (e.g., text in slides are usually shorter).

Intuitively, the number of slides in a deck is a good proxy for the overall text length. If too short, too much text will be put on the same slide, making it difficult to read; conversely, if a deck has too many slides, each slide can convey only little information while making the whole presentation lengthy. Therefore, we propose the slide-level ROUGE:

$$\text{ROUGE-SL} = \text{ROUGE-L} \times e^{\frac{|Q-\tilde{Q}|}{Q}}, \tag{14}$$

where Q and $\tilde{Q}$ are the number of slides in the generated and the ground-truth slide decks, respectively.

Longest Common Figure Subsequence (LC-FS) We measure the quality of figures in the output slides by considering both the correctness (whether the figures from the ground-truth deck are included) and the order (whether all the figures are ordered logically – i.e, in a similar manner to the ground-truth deck). To this end, we use the Longest Common Subsequence (LCS) to compare the list of figures in the output $\{I_0^{out}, I_1^{out}, ...\}$ to the ground-truth $\{\tilde{I}_0^{out}, \tilde{I}_1^{out}, ...\}$ and report precision/recall/F1.

Text-Figure Relevance (TFR) A good slide deck should put text with relevant figures to make the presentation informative and attractive. In addition to considering text and figures independently, we measure the relevance between them. We again adapt ROUGE and modify as

$$\text{TFR} = \frac{1}{M_F^{in}} \sum\nolimits_{i=0}^{M_F^{in}-1} \text{ROUGE-L}(P_i, \tilde{P}_i), \tag{15}$$

where P_i and $\tilde{P}_i$ are sentences from generated and ground-truth slides that contain I_i^{in}, respectively.

Mean Intersection over Union (mIoU) A good design layout makes it easy to consume information presented in slides. To evaluate the layout quality, we adapt the mean intersection over union (mIoU) [24] by incorporating the LCS idea. Given a generated slide deck $\mathcal{O}$ and the ground-truth $\tilde{\mathcal{O}}$, we compute:

$$\text{mIoU}(\mathcal{O}, \tilde{\mathcal{O}}) = \frac{1}{N_O^{out}} \sum\nolimits_{i=0}^{N_O^{out}-1} \text{IoU}(O_i, \tilde{O}_{J_i}) \tag{16}$$

where $\text{IoU}(O_i, \tilde{O}_j)$ computes the IoU between a set of predicted bounding boxes from slide i and a set of ground-truth bounding boxes from slide and J_i. To account for a potential structural mismatch (with missing/extra slides), we find the $J = \{j_0, j_1, ..., j_{N_O^{out}-1}\}$ that achieves the maximum mIoU between $\mathcal{O}$ and $\tilde{\mathcal{O}}$ in an increasing order.

5.2. Implementation Detail

For the DR, we use a Bi-GRU with 1,024 hidden units and set the MLPs to output 1,024-dimensional embeddings. Each layer of the PT is based on a 256-unit GRU. The PAR is designed as an attention-based seq2seq model [7] with 512 hidden units. All the MLPs are two-layer fully-connected networks. We train our network end-to-end using ADAM [21] with a learning rate of 3e-4.

	Ablation Settings				ROUGE-SL		LC-FS			TFR	mIoU
	Hrch-PT	PAR	TIM	Post Proc.	Ours	w/o SL	Prec.	Rec.	F1		(Layout / Template)
(a)	✗	✗	✗	✗	24.35	29.77	25.54	14.85	18.78	5.61	43.34 / 38.15
(b)	✓	✗	✗	✗	24.93	29.68	17.48	26.26	20.99	8.58	49.16 / 40.94
(c)	✓	✓	✗	✗	27.19	32.27	17.48	26.26	20.99	9.23	49.16 / 40.94
(d)	✓	✗	✓	✗	26.52	30.99	23.47	25.31	24.36	10.09	**50.82** / 42.96
(e)	✓	✓	✓	✗	**29.40**	**34.27**	23.47	25.31	24.36	11.82	**50.82** / 42.96
(f)	✓	✓	✓	✓	**29.40**	**34.27**	**26.36**	**38.39**	**31.26**	**17.49**	- / 46.73

Table 2. An overall result of different ablation settings under automatic evaluation metrics ROUGE-SL, LC-FS, TFR, and mIoU.

Train ↓ / Test →	CV	NLP	ML	All
CV	**31.2 / 32.1 / 19.7**	24.1 / 21.5 / 5.6	24.0 / 25.6 / 11.2	24.7 / 29.2 / 15.8
NLP	28.8 / 30.0 / 13.4	**34.7 / 30.7 / 11.8**	29.2 / 32.7 / 15.3	28.9 / 30.9 / 13.6
ML	21.1 / 29.2 / 11.6	21.1 / 26.6 / 6.6	**32.1 / 36.8 / 22.8**	24.9 / **31.4** / 14.4
All	29.2 / 31.2 / 18.6	30.0 / 28.8 / 9.7	29.4 / 32.9 / 20.6	**29.4** / 31.3 / **17.5**

Table 3. Topic-aware evaluation results (ROUGE-SL / LC-F1 / TFR) when trained and tested on data from different topics.

5.3. Results and Discussions

Is the hierarchical modeling effective? To answer this question we define a "flattened" version of our Progress Tracker (flat-PT) by replacing the hierarchical RNN with a vanilla RNN that learns a single shared latent space to model the section-slide-object structure. The flat-PT contains a single GRU and a two-layer MLP with a ternary decision head that learns a policy ζ to predict an action $a_t = \{\texttt{[NEW\_SECTION]}, \texttt{[NEW\_SLIDE]}, \texttt{[NEW\_OBJ]}\}$. For a fair comparison, we increase the number of hidden units in the baseline GRU to 512 (ours is 256) so the model capacities are roughly the same between the two.

First, we compare the structural similarity between the generated and the ground-truth slide decks. For this, we build a list of tokens indicating a section-slide-object structure (e.g., [SEC], [SLIDE], [OBJ], ..., [SLIDE], ...) and compare the lists using the longest common subsequence (LCS). Our hierarchical approach achieves 64.15% vs. the flat approach 51.72%, suggesting that ours was able to learn the structure better than baseline.

Table 2 (a) and (b) compare the two models on the four metrics introduced in Sec. 5.1. The results show that ours outperforms flat-PT across all metrics. The flat-PT achieves slightly better performance on ROUGE-SL without the slide-length term (w/o SL), which is the same as ROUGE-L. This suggests that ours generates a slide structure more similar to the ground-truth than the flat approach.

A deeper look into the content similarity loss We ablate different terms in the content similarity loss (Eq. 12) to understand their individual effectiveness; shown in Table 2.

PAR. The paraphrasing loss improves text quality in slides; see the ROUGE-SL scores of (b) vs. (c), and (d) vs. (e). It also improves the TFR metric because any improvement in text quality will benefit text-image relevance.

TIM. The text-image matching loss improves the figure quality; see (b) vs. (d) and (c) vs. (e). It particularly improves LC-FS precision with a moderate drop in recall rate, indicating the model added more correct figures. TIM also improves ROUGE-SL because it helps constrain the multimodal embedding space, resulting in better selection of text.

Figure post-processing At test time, we leverage the multimodal projection head $\delta(\cdot)$ as a post-processing module to add missing figures and/or remove unnecessary ones. Table 2 (f) shows this post-processing further improves the two image-related metrics, LC-FS and TFR. For simplicity, we add figures following equally fitting in template-based design instead of using OP to predict its location.

Layout prediction vs. templates The object placer (OP) predicts the layout to decide where and how to put the extracted objects. We compare this with a template-based approach, which selects the current section title as the slide title and puts sentences and figures in the body line-by-line. For those extracted figures, they will equally fit (with the same width) in the remaining space under the main content.

The result shows that the predicted-based layout, which directly learns from the layout loss, can bring out higher mIoU with the groundtruth. But for the template-based design, in the aspect of the visualization, it can make the generated slide deck more consistent.

Topic-aware evaluation We evaluate performance in a topic-dependent and independent fashion. To do this, we train and test our model on data from each of the three research communities (CV, NLP, and ML). Table 3 shows that models trained and tested within each topic performs the best (not surprisingly), and that models trained on data from all topics achieves the second best performance, showing generalization to different topic areas. Training on NLP data, despite being the smallest among the three, seems to

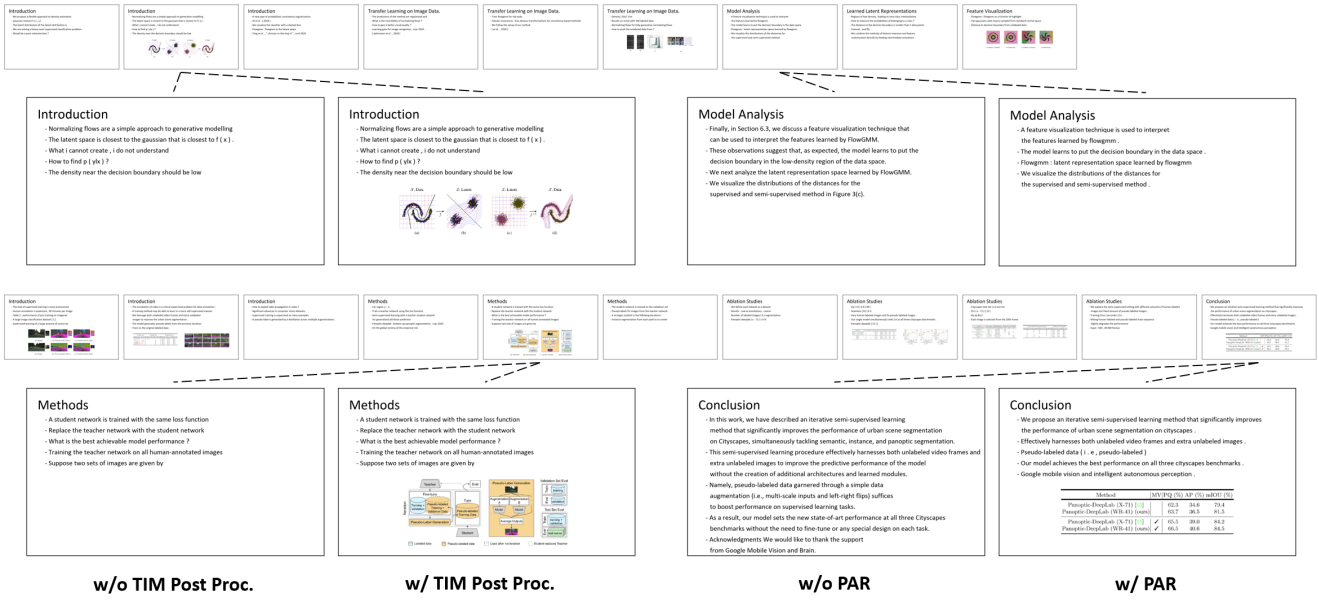


Figure 3. Qualitative examples of the generated slide deck from our model (Paper source: top [33] and bottom [11]). We provide more results, including failure cases, on our project webpage.

generalize well to other topics on the text metric, achieving the second best on ROUGE-SL (28.9). Training on CV data provides the second highest performance on the text-figure metric TFR (15.8), and training on ML achieves the highest figure extraction performance (LC-FS F1 of 31.4).

Human evaluation We conduct a user study to assess the perceived quality of generates slides. We select 50 documents from the test set and prepare four slide decks per document: the ground-truth deck, and the ones generated by the flat PT (Table 2 (a)), by ours without PAR and TIM (b), and by our final model (f). To make the task easy to complete, we sample 200 sections from 50 documents and create 600 pairs of ground-truth and generate slides.

We recruited three AMT Master Workers for each task (HIT). The workers were shown the slides from the ground-truth deck (DECK A) and one of the methods (DECK B). The workers were then asked to answer three questions: `Q1`. Looking only at the TEXT on the slides, how similar is the content on the slides in DECK A to the content on the slides in DECK B?; `Q2`. How well do the figure(s)/tables(s) in DECK A match the text or figures/tables in DECK B?; `Q3`. How well do the figure(s)/table(s) in DECK A match the TEXT in DECK B? The responses were all on a scale of 1 (not similar at all) to 7 (very similar). Fig. 4 shows the average scores for each method. The average rating for our approach was significantly greater for all three questions compared to the other two methods. There was no significant difference between the ratings for the other two methods.

Qualitative Results Fig. 3 illustrates two qualitative examples (top [33] and bottom [11]) of the slide deck generated by our model. With the post-processing, TIM can add the related figure into the slide and make it more informative. PAR helps create a better presentation by paraphrasing the sentences into bullet point form.

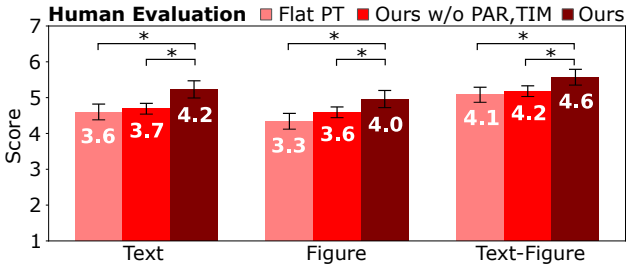


Figure 4. The average scores for how closely the generated text and figures match the text and figures in the ground-truth slides, respectively. And how well the generated text matches the figures in the ground-truth slides. Error bars reflect standard error. Significance tests: two-sample t-test (p <0.05.)

6. Conclusion

We present a novel task and approach for generating slides from documents. This is a challenging multimodal task that involves understanding and summarizing documents containing text and figures and structuring it into a presentation form. We release a large set of 5,873 paired documents and slide decks, and provide evaluation metrics with our results. We hope our work will help advance the state-of-the-art in vision and language understanding.

A. Details of the Data Processing Steps

Section 4 in our main paper explains how we construct our DOC2PPT dataset. Here we provide the details of the process and demonstrate the accuracy of the various extraction/matching processes. Fig. 5 illustrates the details of the data processing pipeline that were omitted in the main paper. To evaluate how reliable the various steps in our pipeline are, we manually labeled 100 slide decks (randomly sampled from the validation split) and used them for evaluation.

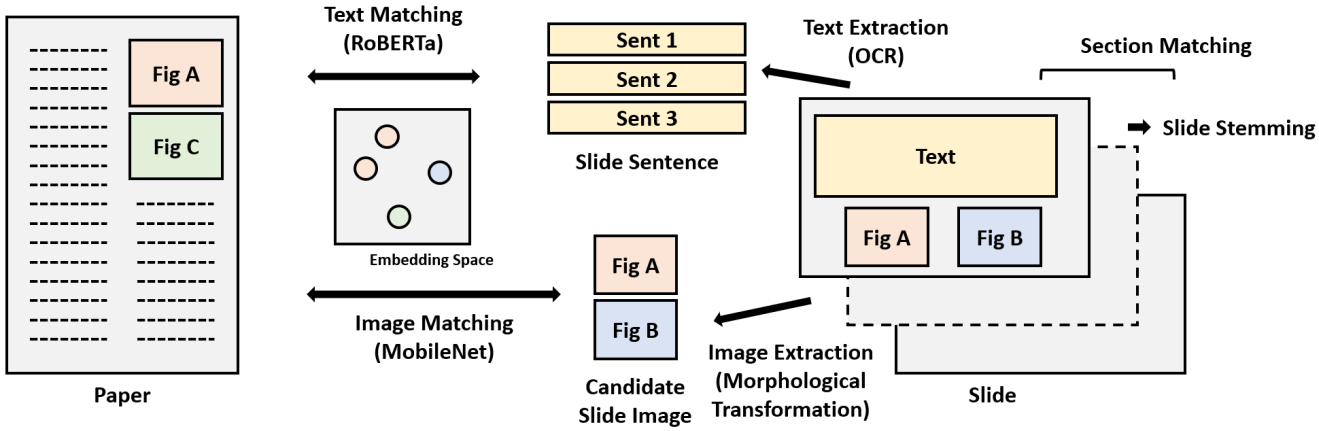


Figure 5. **Data processing pipeline.** We automatically extract text/figures and match them between each document and slide deck pair.

Text Extraction Fig. 6 shows examples of the extracted slide sentences obtained using Azure OCR [1]. The original slides are shown on left and the extracted text is on the right. Notice that the OCR results are quite reliable as slides contain text in a clear format.

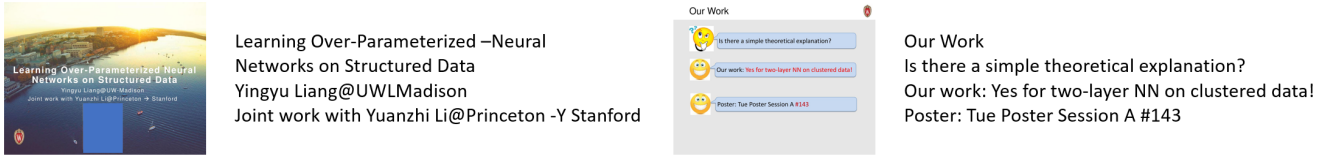


Figure 6. **Text Extraction from Slide Deck.** We use Azure OCR [1] to extract sentences from slides.

Slide Stemming Fig. 7 illustrates the slide stemming process. If a slide has a preceding slide with 80% or greater overlap in content, we consider the preceding slide as redundant and remove it. The slides which are opaque (ghosted) are examples of slides that would be removed (they often exist because of animations that sequentially add elements to a slide - e.g., bullet points appearing - thus we just keep the final slide in the sequence to simplify the dataset). Our slide stemming step is 93% accurate based on the human annotated validation set.

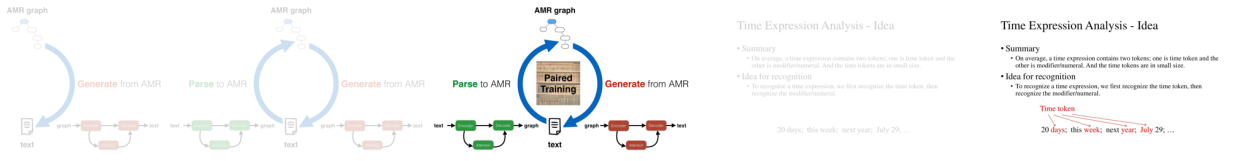


Figure 7. **Slide Stemming.** The ghosted/opaque slides are seen as redundant and will be removed by the stemming process. This helps simplify our dataset.

Slide-Section Matching Fig. 8 presents an example of slide-section matching. We adopt RoBERTa [46] to extract embeddings of the text in slides and sections in the document (paper). Specifically, we find slide-to-section matching based on the cosine similarity between text embeddings. Slides are matched with the section with the highest cosine similarity and our slide-section matching has 82% accuracy.

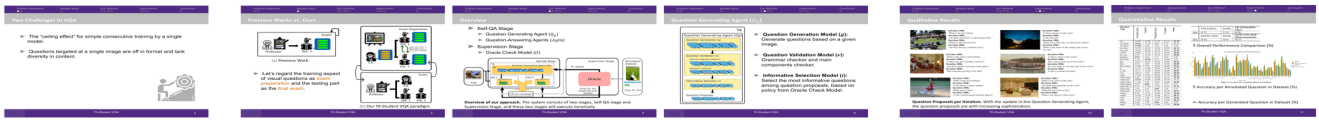


Figure 8. **Slide-Section Matching.** We match slides to the corresponding sections in the document so that a slide deck is represented as a set of non-overlapping section groups.

Sentence Matching Table 4 shows examples of matching sentences between the paper and the slide. We again use RoBERTa [46] to search for the matching sentence based on the cosine similarity and build the linking for the extractive summarization.

Paper Sentence	Slide Sentence
The Pima Indians Diabetes data set contains information about 768 diabetes patients, recording features like glucose blood, pressure, age, and skin thickness	This data set contains 768 diabetes patients, recording features like glucose, blood
Finally, can the idea of proportionality as a group fairness concept be adapted for supervised learning tasks like classification and regression	Can fairness as proportionality be adapted for supervised

Table 4. **Sentence Matching.** The example of matching sentences from the slide to the paper.

Figure Matching Fig. 9 illustrates examples of figures/tables that were matched with a particular slide. We apply morphological transformation [2] and border following [57] to extract possible slide figures. We then match them with figures in the paper using the visual embedding from MobileNet [31]; if the cosine similarity is larger than the threshold θ_I. Fig. 10 presents the precision, recall, and F1, which are evaluated from human-labeled test set. The x-axis represents different values of threshold θ^I considered when comparing the cosine similarity of the visual embedding. When θ^I is lower, more figures from the paper will be included, which increases recall but negatively impacts precision; in contrast, a higher θ^I results in greater precision but lower recall. Fig. 11 shows examples where the figure matching performs poorly. There are two cases: 1) partial figure matches where a figure has had elements added or removed, and 2) different versions of a figure where the meaning might be similar but the images do not match. These cases make matching difficult, because based on the visual embedding they may not be very similar.

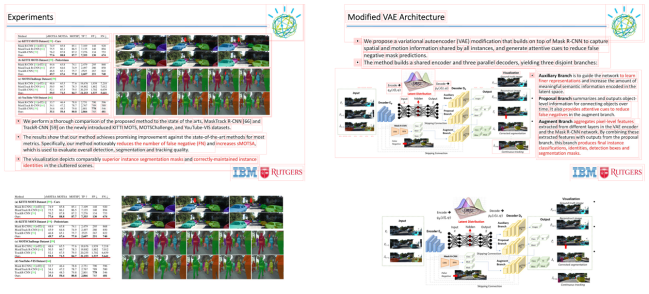


Figure 9. **Figure Matching.** The lower figures are those matched from the paper using the cosine similarity and features from MobileNet [31].

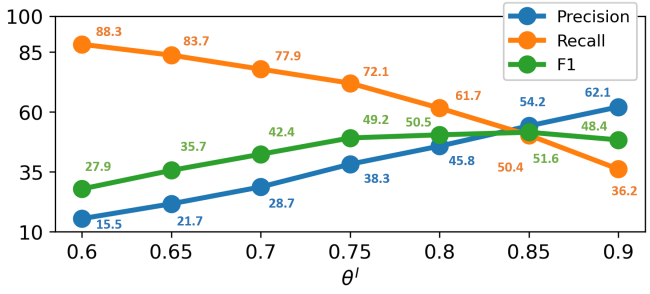
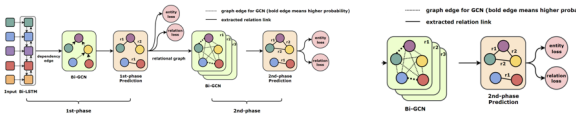


Figure 10. **Figure Matching under Different θ^I.** Precision, recall, and F1 are evaluated using the human-labeled testing set.



Partial Matching

Method	NYT			WebNLG		
	Precision	Recall	F1	Precision	Recall	F1
NovelTagging	62.4%	31.7%	42.0%	**52.5%**	19.3%	28.3%
OneDecoder	59.4%	53.1%	56.0%	32.2%	28.9%	30.5%
MultiDecoder	61.0%	56.6%	58.7%	37.7%	36.4%	37.1%
GraphRel$_{1p}$	62.9%	57.3%	60.0%	42.3%	39.2%	40.7%
GraphRel$_{2p}$	**63.9%**	**60.0%**	**61.9%**	44.7%	**41.1%**	**42.9%**

P	R	F1	NER
52.5%	19.3%	28.3%	-
37.7%	36.4%	37.1%	-
42.3%	39.2%	40.7%	89.1%
44.7%	**41.1%**	**42.9%**	**91.9%**

Different Expression

Figure 11. **Partial Matching and Different Expression.** The examples where the figure matching performs poorly.

Human Labeling To ensure that our test set is clean and reliable, we use Amazon Mechanical Turk (AMT) and have humans perform image extraction and matching for the entire testing set. Fig. 12 shows a screenshot of the MTurk HIT for labeling figure matches within each slide. The slide is shown on the left and figures from the document (paper) were shown on the right. The human annotators can label each figure either as a match (by clicking on the image) or as similar but not an exact match (by ticking the checkbox next to the image). Fig. 13 shows a screenshot of the MTurk HIT for labeling the bounding box around the image on a slide. The candidate figure is shown above and the human annotator is asked to draw a bounding box around the region of the slide where it appeared. We perform figure-slide matching (see above) before bounding box labeling as this produced the best quality annotations (bounding box labeling is not necessary if the image isn't on the slide at all). For the human-labeled testing set, a slide deck contains on average 2.3 images that are excerpted from the corresponding paper. Please note that since people tend to adopt more new figures or different figures in a slide deck for computer vision (CV) field, the average number of excerpted figure is lower (1.7).

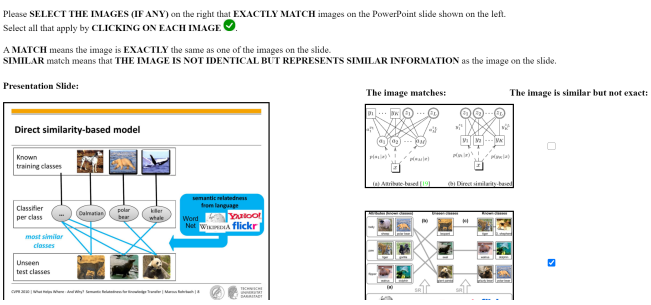


Figure 12. **Interface of Figure Matching Labeling.** The annotator label figures either as match or as similar but not exact match.

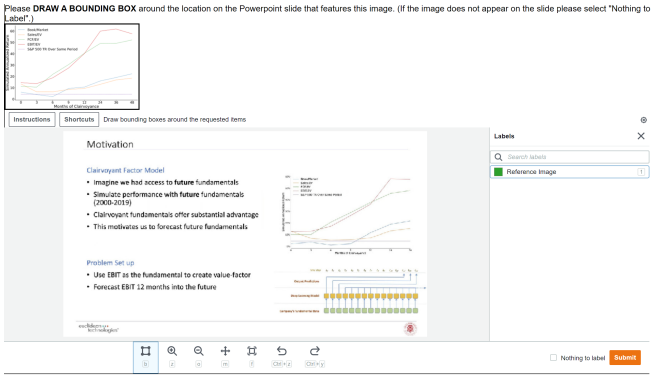


Figure 13. **Interface of Bounding Box Labeling.** The annotator is asked to draw a bounding box around the region of the slide where the candidate figure appeared.

B. Settings of Approach

The importance of h^{obj} in Paraphrasing Module Table 5 presents the Rouge-L of paraphrasing module (PAR) with or without using the object state h^{obj}. The results show that the text quality improves in all cases if we apply PAR. Also, using h^{obj} benefits more (w/ 32.27 vs w/o 31.95). This is because h^{obj} provides contextual information, which helps PAR generate a paraphrased sentence more relevant to the content in the document.

Sensitivity of θ^R and θ^A in Post-Processing During the post-processing, we remove figures deemed irrelevant by θ^R and add ones if considered highly relevant based on θ^A. To achieve the best result, we tune our θ^R and θ^A on the 100 labeled validation set. Fig. 14 shows that $\theta^R = 0.8$ and $\theta^A = 0.9$ achieves the highest LC-F1.

C. Human Evaluation

Fig. 15 shows a screenshot of the human rating task for evaluating the quality of the generated slides. The ground-truth slide deck was shown (left) alongside the generated slides (right). The human annotators were asked three questions. 1) How similar the text on slide DECK A was to the text on slide DECK B. 2) How similar the figures on slide DECK A was to the figures on slide DECK B - the could also indicate that no figures were present. 3) How similar the figures in DECK B were to the text in DECK A - again they could indicate that no figures were present if that was the case.

Ablation Settings			Rouge-L	
Hrch-PT	TIM	PAR	w/o h^{obj}	w/ h^{obj}
✓	✗	✗	29.68	
✓	✗	✓	31.95	**32.27**
✓	✓	✗	30.99	
✓	✓	✓	33.05	**34.27**

Table 5. **Considering h^{pbj} in PAR.** The Rouge-L score of with and without h^{obj} in paraphrasing module.

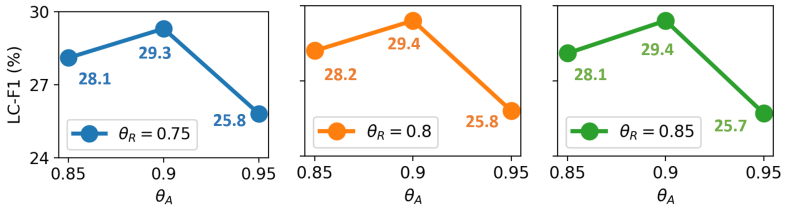


Figure 14. **Post-Processing under different θ^R and θ^A.** We tune the θ^R and θ^A for the post-processing based on the LC-F1 on the validation set.

D. Qualitative Examples

Fig. 17 demonstrates generated slide decks from our approach. We provide more results, including failure cases, on our project webpage.

Applying PowerPoint Design Ideas As we discussed in the main paper, the output of our method can be used as a draft slide deck for humans to build upon. We provide one such application scenario of our approach. When the slide decks are generated based on a template, the content are all in a fixed size and in the fix position. To make the output more attractive, we can apply off-the-shelf tools such as Microsoft PowerPoint Design Ideas [3] which can automatically produce a layout for the given texts and figures. As shown in Fig. 16, the generated decks are more professional looking.

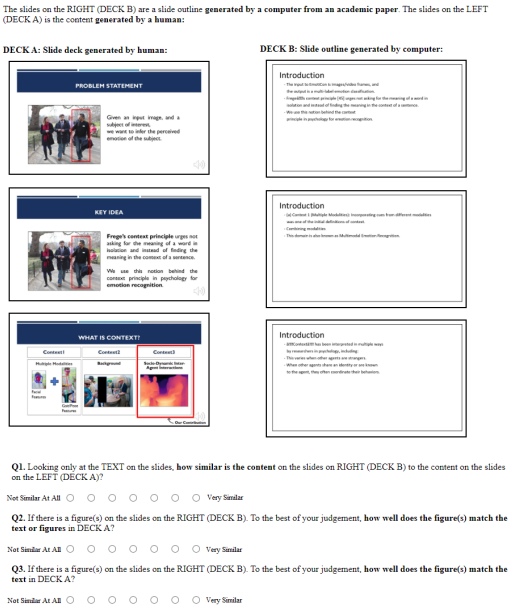


Figure 15. **Interface of Human Evaluation.** The annotator is asked three questions on aspects of the text quality, the figure extraction, and the text-figure relevance.

Figure 16. **Applying PowerPoint Design Ideas.** By applying the design ideas feature [3] provided from Microsoft PowerPoint, we can make the generated template-based slide deck more professional and more attractive for the presentation.

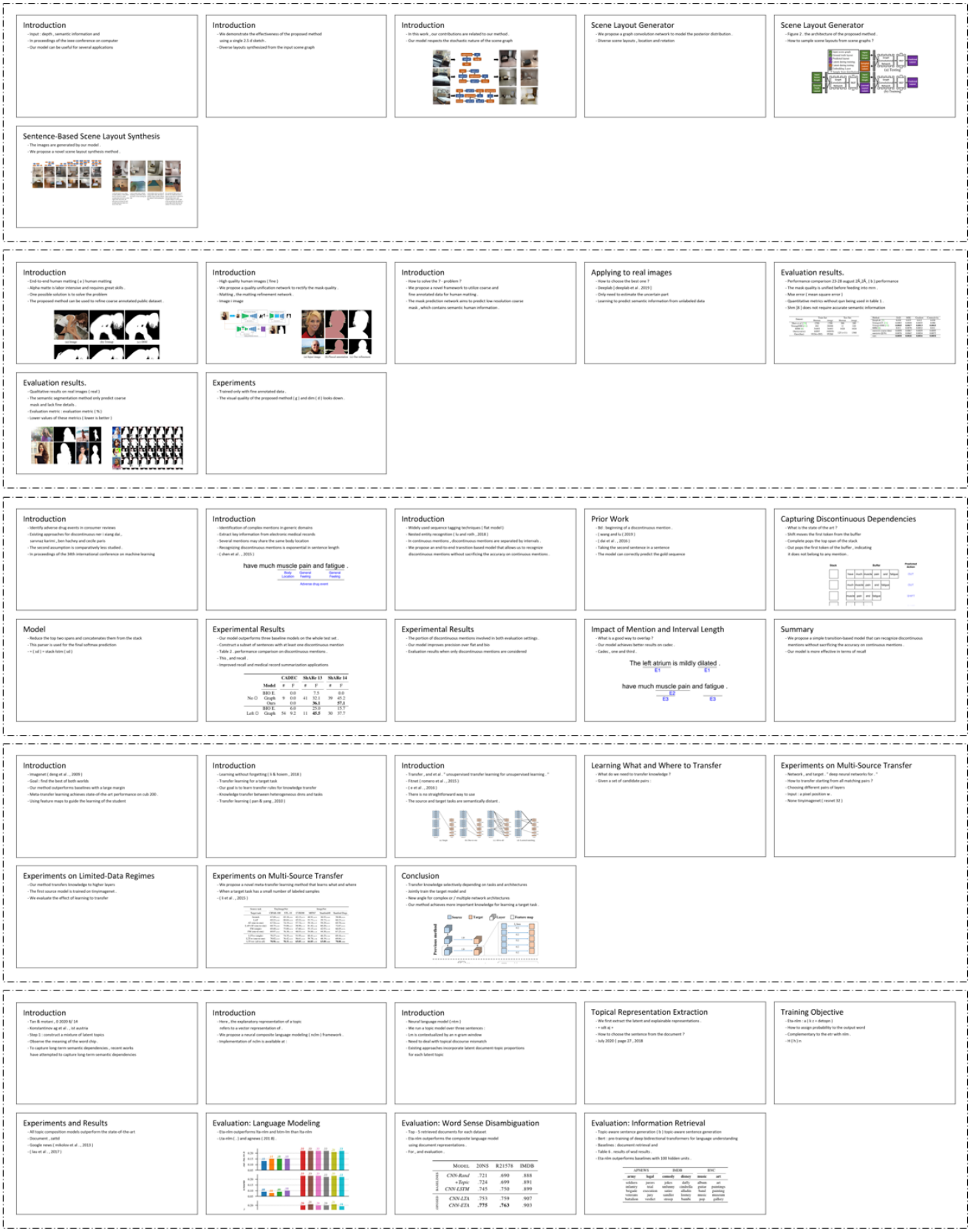


Figure 17. **Qualitative examples.** (from top to bottom: [48], [42], [19], [35], and [10]) Please visit our webpage for more generated slide decks from our approach.

References

[1] *Azure Cognitive Services*. https://docs.microsoft.com/en-us/azure/cognitive-services. 5, 9

[2] *OpenCV*. https://opencv.org/. 5, 10

[3] *PowerPoint Design Ideas*. https://reurl.cc/gmjm87. 12

[4] *ScienceParse*. https://github.com/allenai/science-parse. 5

[5] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[6] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR)*, 2015. 4, 6

[8] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. Variations of the Similarity Function of TextRank for Automated Summarization. In *Argentine Symposium on Artificial Intelligence (ASAI)*, 2015. 2

[9] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep Communicating Agents for Abstractive Summarization. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018. 2

[10] Yatin Chaudhary, Hinrich Schütze, and Pankaj Gupta. Explainable and Discourse Topic-aware Neural Language Understanding. In *International Conference of Machine Learning (ICML)*, 2020. 13

[11] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D. Collins, Ekin D. Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Semi-Supervised Learning in Video Sequences for Urban Scene Segmentation. In *European Conference on Computer Vision (ECCV)*, 2020. 8

[12] Xiuying Chen, Shen Gao, Chongyang Tao, Yan Song, Dongyan Zhao, and Rui Yan. Iterative Document Representation Learning Towards Summarization with Polishing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 2

[13] Jianpeng Cheng and Mirella Lapata. Neural Summarization by Extracting Sentences and Words. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016. 2

[14] Jaemin Cho, Minjoon Seo, and Hannaneh Hajishirzi. Mixture Content Selection for Diverse Sequence Generation. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. 2

[15] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 4

[16] Sumit Chopra, Michael Auli, and Alexander M Rush. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2016. 2

[17] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *Advances in Neural Information Processing Systems Workshop (NeurIPS WS)*, 2014. 3

[18] Christopher Clark and Santosh Divvala. PDFFigures 2.0: Mining Figures from Research Papers. In *IEEE/ACM Joint Conference on Digital Libraries(JCDL)*, 2016. 5

[19] Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. An Effective Transition-based Model for Discontinuous NER. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020. 13

[20] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. Visual Dialog. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[21] Jimmy Ba Diederik P. Kingma. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2014. 6

[22] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[23] Aaron Elkiss, Siwei Shen, Anthony Fader, Güneş Erkan, David States, and Dragomir Radkov. Blind Men and Elephants: What do Citation Summaries Tell Us about a Research Article? In *Journal of the American Society for Information Science and Technology (JASIST)*, 2008. 2

[24] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. In *International Journal of Computer Vision (IJCV)*, 2010. 6

[25] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *British Machine Vision Conference (BMVC)*, 2018. 2

[26] Andrea Frome, Greg S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. DeViSE: A Deep Visual-Semantic Embedding Model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013. 2, 5

[27] Jiuxiang Gu, Jianfei Cai, Shafiq Joty, Li Niu, and Gang Wang. Look, Imagine and Match: Improving Textual-Visual Cross-Modal Retrieval with Generative Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[28] Jiatao Gu, Zhengdong Lu, Hang Li, Victor O.K, and Li. Incorporating Copying Mechanism in Sequence-to-Sequence

Learning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016. 4

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015. 5

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[31] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6, 10

[32] Yan Huang, Qi Wu, and Liang Wang. Learning Semantic Concepts and Order for Image and Sentence Matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[33] Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-Supervised Learning with Normalizing Flows. In *International Conference of Machine Learning (ICML)*, 2020. 8

[34] Kokil Jaidka, Muthu Kumar, Chandrase karan, and Sajal Rustagi amd Min-Yen Kan. Overview of the CL-SciSumm 2016 Shared Task. In *Joint Workshop on Bibliometric-enhanced Information Retrieval and NLP for Digital Libraries (BIRNDL)*, 2016. 2

[35] Yunhun Jang, Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Learning What and Where to Transfer. In *International Conference of Machine Learning (ICML)*, 2019. 13

[36] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. VQA: Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[37] Andrej Karpathy and Li Fei-Fei. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 5

[38] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. In *Advances in Neural Information Processing Systems Workshop (NeurIPS WS)*, 2014. 2, 5

[39] Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. TalkSumm: A Dataset and Scalable Annotation Method for Scientific Paper Summarization Based on Conference Talks. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. 2

[40] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[41] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014. 6

[42] Jinlin Liu, Yuan Yao, Wendi Hou, Miaomiao Cui, Xuansong Xie, Changshui Zhang, and Xian sheng Hua. Boosting Semantic Human Matting with Coarse Annotations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 13

[43] Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. Generative Adversarial Network for Abstractive Text Summarization. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018. 2

[44] Yang Liu. Fine-tune BERT for Extractive Summarization. In *arXiv:1903.10318*, 2019. 2

[45] Yang Liu and Mirella Lapata. Text Summarization with Pretrained Encoders. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. 2

[46] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In *arxiv:1907.11692*, 2019. 3, 6, 10

[47] Elena Lloret, María Teresa Romá-Ferri, and Manuel Palomar. COMPENDIUM: A Text Summarization System for Generating Abstracts of Research Papers. In *Data & Knowledge Engineering*, 2013. 2

[48] Andrew Luo, Zhoutong Zhang, Jiajun Wu, and Joshua B. Tenenbaum. End-to-End Optimization of Scene Layout. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 13

[49] Luca Marchesotti, Florent Perronnin, Diane Larlus, and Gabriela Csurka. VMSMO: Learning to Generate Multimodal Summary for Video-based News Articles. In *International Conference on Computer Vision (ICCV)*, 2011. 2

[50] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018. 2

[51] Daraksha Parveen, Mohsen Mesgar, and Michael Strube. Generating Coherent Summaries of Scientific Articles Using Coherence Patterns. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016. 2

[52] Romain Paulus, Caiming Xiong, and Richard Socher. A Deep Reinforced Model for Abstractive Summarization. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[53] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 5

[54] Alexander M. Rush, Sumit Chopra, and Jason Weston. A Neural Attention Model for Abstractive Sentence Summarization. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015. 2

[55] Abigail See, Peter J. Liu, and Christopher D. Manning. Get To The Point: Summarization with Pointer-Generator Networks. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017. 2

[56] Yale Song and Mohammad Soleymani. Polysemous Visual-Semantic Embedding for Cross-Modal Retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[57] Satoshi Suzuki and Keiichi Abe. Topological Structural Analysis of Digitized Images by Border Following. In *Computer Vision, Graphics, and Image Processing (CVGIP)*, 1985. 5, 10

[58] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-Embeddings of Images and Language. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[59] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016. 2

[60] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[61] Ronald J. Williams and David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. In *Neural computation*, 1989. 5

[62] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[63] Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. ScisummNet: A Large Annotated Corpus and Content-Impact Models for Scientific Paper Summarization with Citation Networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2019. 2

[64] Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. Graph-based Neural Multi-Document Summarization. In *Conference on Computational Natural Language Learning (CoNLL)*, 2017. 2

[65] Wenpeng Yin and Yulong Pei. Optimizing Sentence Modeling and Selection for Document Summarization. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2014. 2

[66] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image Captioning with Semantic Attention. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[67] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *International Conference of Machine Learning (ICML)*, 2020. 2

[68] Junnan Zhu, Haoran Li, Tianshang Liu, Yu Zhou, Jiajun Zhang, and Chengqing Zong. MSMO: Multimodal Summarization with Multimodal Output. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 2

[69] Junnan Zhu, Yu Zhou, Jiajun Zhang, Haoran Li, Chengqing Zong, and Changliang Li. Multimodal Summarization with Guidance of Multimodal Reference. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020. 2