SlideVQA: A Dataset for Document Visual Question Answering on Multiple Images

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, Kuniko Saito

NTT Human Informatics Laboratories, NTT Corporation
{ryouta.tanaka.rg, kyosuke.nishida.rx, kosuke.nishida.ap, taku.hasegawa.ps, itsumi.saito.df, kuniko.saito.ku}@hco.ntt.co.jp

Abstract

Visual question answering on document images that contain textual, visual, and layout information, called document VQA, has received much attention recently. Although many datasets have been proposed for developing document VQA systems, most of the existing datasets focus on understanding the content relationships within a single image and not across multiple images. In this study, we propose a new multi-image document VQA dataset, SlideVQA, containing 2.6k+ slide decks composed of 52k+ slide images and 14.5k questions about a slide deck. SlideVQA requires complex reasoning, including single-hop, multi-hop, and numerical reasoning, and also provides annotated arithmetic expressions of numerical answers for enhancing the ability of numerical reasoning. Moreover, we developed a new end-to-end document VQA model that treats evidence selection and question answering in a unified sequence-to-sequence format. Experiments on SlideVQA show that our model outperformed existing state-of-the-art QA models, but that it still has a large gap behind human performance. We believe that our dataset will facilitate research on document VQA.

Introduction

Building intelligent agents that can read and comprehend real-world documents, such as webpages, office documents, lecture slides, etc., has been a long-standing goal of artificial intelligence. To achieve this goal, machine reading comprehension (MRC), a central task in natural language understanding, has been intensively studied. The typical definition of the MRC task is quite simple, wherein given a short natural language text as a context and a question about it, a machine reads the text and then answers the question by extracting a span from the text (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018). However, this definition is far from real-world applications, such as customer service chatbots on e-commerce websites (Cui et al. 2017) and assistant systems for reading professional literature (Hong et al. 2019), in that the context is composed entirely of text, with no graphical elements.

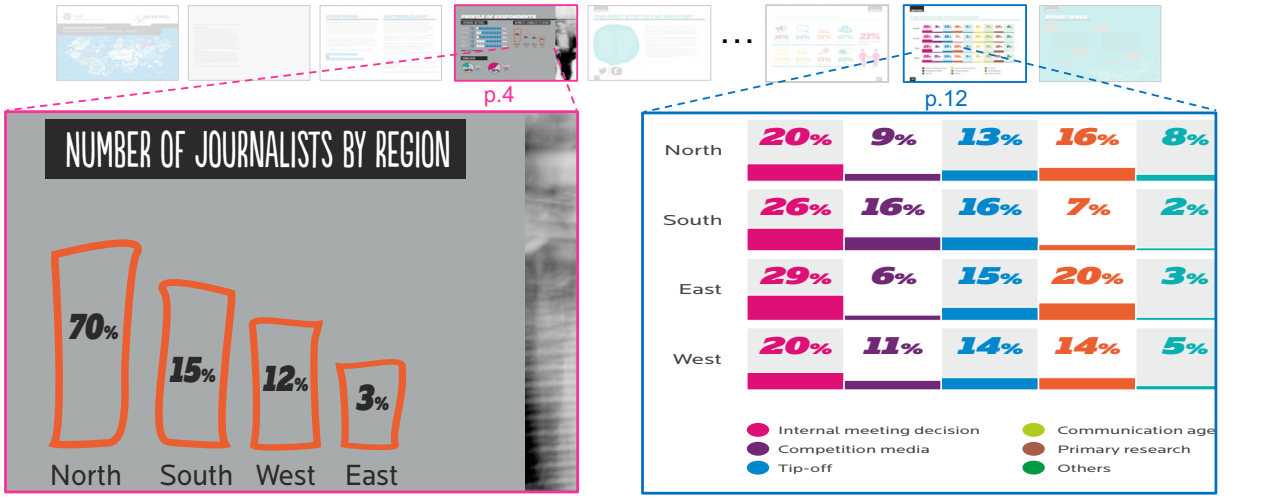
To this end, visual question answering on document images (document VQA) has received much attention. It is a challenging vision and language task that requires methods to reason about document layout, textual content, and visual elements (Mathew, Karatzas, and Jawahar 2021; Tanaka, Nishida, and Yoshida 2021; Mathew et al. 2022). When the primary content in a document is text (e.g., e-mails and forms) and the task is to understand it on the basis of its layout information, state-of-the-art models have already achieved nearly human-level performance (Xu et al. 2021; Powalski et al. 2021). On the other hand, challenges remain when it comes to handling diverse real-world documents. First and foremost is that current models are not capable of performing reasoning across multiple images since the existing datasets focus on testing reasoning ability on a single image. Moreover, compared with humans, document VQA models still have trouble understanding documents that contain visual elements and understanding questions that require numerical reasoning (Mathew et al. 2022).

To address the above challenges, we introduce a new document VQA dataset[1], SlideVQA, for tasks wherein given a slide deck composed of multiple slide images and a corresponding question, a system selects a set of evidence images and answers the question. Slide decks are one of the most efficient document types that arrange visual and textual elements for communication. As shown in Figure 1, SlideVQA requires complex reasoning over slide images, including single-hop, multi-hop, and numerical reasoning. These reasoning skills play essential roles in MRC tasks (Yang et al. 2018; Dua et al. 2019).

Our main contributions are summarized as follows:

- We introduce a novel task and dataset, SlideVQA, wherein to answer its questions, a machine has to read and comprehend a slide deck composed of multiple images. It is the largest multi-image document VQA dataset containing 2.6k+ slide decks (each consisting of 20 slides) and 14.5k questions. It also provides bounding boxes around textual and visual elements for understanding document layout and arithmetic expressions for numerical reasoning.
- We developed a **M**ulti-**M**odal **M**ulti-image **D**ocument VQA model, M3D, to jointly perform evidence selection and question answering tasks and to enhance numerical reasoning by generating arithmetic expressions.

[1]Our dataset and codes are publicly available at https://github.com/nttmdlab-nlp/SlideVQA



Q: What is the tip-off media percentage in the region with 70% of journalists and South?

Answer: 13%, 16%
Answer type: Multi-Span
Evidence pages: 4, 12
Reasoning type: Multi-hop

Q: What is the difference in the competition media percent age between East and the region with 12% of journalists?

Answer: 5% (11% – 6%)
Answer type: Non-Span
Evidence pages: 4, 12
Reasoning type: Multi-hop, Numerical

Figure 1: Examples from our SlideVQA dataset. Some questions can be answered through single-hop, multi-hop, and numerical reasoning. The colors of the words match the image borders with the same colors. (·) of the right example in the answer denotes an annotated arithmetic expression to derive the final answer. The slide deck can be viewed at https://www.slideshare.net/mslgroup/mediainsights-evolving-sources-of-news-for-media.

- Our model outperformed existing state-of-the-art QA models on SlideVQA, but its performance is still below that of humans by a large margin.

Related Work

Datasets for VQA on document images. Document VQA is the task of answering questions about document images, and some useful datasets have been published, such as DocVQA (Mathew, Karatzas, and Jawahar 2021), VisualMRC (Tanaka, Nishida, and Yoshida 2021), WebSRC (Chen et al. 2021), and InfographicVQA (Mathew et al. 2022). The task assumes that the datasets have a single relevant image, containing all the facts required to answer.

The work most related to ours is DocCVQA (Tito, Karatzas, and Valveny 2021), wherein a large collection of document images is used to answer a given question. Our dataset differs from DocCVQA, as follows. First, SlideVQA consists of 14.5k questions, wheres DocCVQA provides only 20 questions. Second, SlideVQA requires multi-hop reasoning over multiple slides to find the answer, while DocCVQA requires only single-hop reasoning on individual images to find the answer. Besides these differences, SlideVQA provides questions that require numerical reasoning and arithmetic expression annotations to answer numerical questions (e.g., "30 - 28" for the answer "2"): no other VQA dataset, including InfographicVQA that requires numerical reasoning, provides such annotations. Furthermore, SlideVQA provides the largest number of bounding boxes on all of the collected images among the related datasets.

Document VQA Models. In parallel with the development of datasets, Transformer (Vaswani et al. 2017) has come to be used for understanding unstructured text in document images. LayoutLM (Xu et al. 2020), LayoutLMv2 (Xu et al. 2021), LayoutT5 (Tanaka, Nishida, and Yoshida 2021), and TILT (Powalski et al. 2021) have achieved impressive results in single-image document VQA tasks by combining textual, layout, and visual features. By contrast, we focus on endowing models with the ability to reason and comprehend multiple images. Moreover, while Tito, Karatzas, and Valveny (2021) used a pipeline of retrieval and reading models for DocCVQA, we use multi-task learning that jointly performs evidence selection and question answering.

Multi-modal question answering. This type takes textual and visual information as input contexts, which is different from document VQA that takes only a document image as the input context. TQA (Kembhavi et al. 2017) is comprised of middle-school science lessons containing diagrams and text. MultiModalQA (Talmor et al. 2021) requires joint reasoning over text, tables, and images in Wikipedia. The motivation behind these studies is similar to ours, but their input is well-formed for machines, and the visual information in the text such as the document layout is dropped from the text in these datasets.

VQA on videos or image sets. VideoQA focuses on answering questions about video frames of TV shows (Lei et al. 2018, 2020) and movies (Tapaswi et al. 2016). A similar task is VQA on image sets (ISVQA), which involves handling photos taken from different viewpoint indoors (Bansal,

Dataset	Document source	Multi-images input	Multi-hop reasoning	Numerical reasoning	Answer type	#QAs	#Images	#BBoxes	#Arithmetic annotations	#Evidence candidates
DocVQA	industry				SS	50k	12k	–	–	1
VisualMRC	web-pages				Ab	30k	10k	64k	–	1
WebSRC	web-pages				SS	400k	6.4k	–	–	1
InfographicVQA	infographics			✓	SS, MS, NS	30k	5k	–	–	1
DocCVQA	industry	✓			MS	0.02k	14k	–	–	14k
SlideVQA (Ours)	slide decks	✓	✓	✓	SS, MS, NS	14.5k	52k	890k	1.7k	20

Table 1: Comparison of question answering datasets on document images. Answer types can be broken down into abstractive (Ab), single-span (SS), multi-span (MS), and non-span (NS).

Zhang, and Chellappa 2020). By contrast, our dataset also requires a model to understand the text in images.

Slide images understanding. Monica Haurilet and Stiefelhagen (2019); Haurilet et al. (2019) introduced a benchmark for object segmentation on slide pages. Sun et al. (2021); Fu et al. (2022) tackled the task of generating slides from research papers. Our work is the first to focus on answering questions on sets of slide images.

Reasoning over textual documents. Numerical reasoning plays an important role in NLP tasks (Dua et al. 2019; Zhang et al. 2020, 2021). Moreover, multi-hop reasoning has taken the spotlight as it aligns with the multi-hop nature of how humans reason to acquire knowledge, and has led to a proliferation of benchmarks (Talmor and Berant 2018; Yang et al. 2018). However, there is as yet no dataset for developing models to perform both multi-hop and numerical reasoning on document images.

The SlideVQA Task and Dataset

Task Overview and Formulation

The SlideVQA task, requires a system to answer a question about a slide deck, which is composed of an ordered set of slide images and to select evidence slide images. We formulate the end-to-end SlideVQA task as follows:

MAINTASK (SlideVQA). Given a question q and a slide deck $\mathbf{I} = \{I_1, \ldots, I_K\}$ ($K = 20$), a model outputs an answer y and selects relevant slides $\hat{\mathbf{I}} = \{\hat{I}_1, \ldots, \hat{I}_{K'}\}$.

The task can be decomposed into two subtasks:

SUBTASK 1 (Evidence Selection). Given a question q and a slide deck $\mathbf{I}$, a model identifies the images $\hat{\mathbf{I}}$ from which to derive the answer y.

SUBTASK 2 (Question Answering). Given a question q and the slide images ($\mathbf{I}$ or $\hat{\mathbf{I}}$), a model outputs an answer y.

SlideVQA has three answer types (see the examples in Figure 1). A single-span answer is a contiguous sequence of tokens in the reading order extracted from the image, and a multi-span answer is formed from multiple spans from the image. A non-span answer is not extracted and is composed of numerical values and visual appearances.

We can also use annotations of bounding boxes around the objects (and their categories) to understand the semantic structure of images and annotations of arithmetic expressions to understand numerical reasoning as additional input at training. These annotations are not given at inference.

Dataset Collection

In this section, we describe the collection process of the SlideVQA dataset. To control the annotation quality, we recruited crowd workers located in English-speaking countries and who had passed a rigorous qualification procedure. Additionally, we asked other workers to assess the quality of the annotated samples after each collection step.

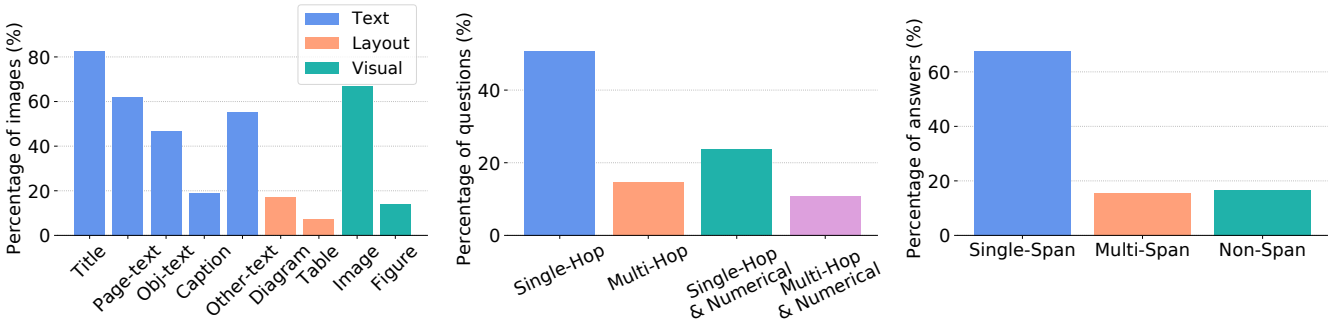
Slide decks collection. First, we selected and downloaded 25,327 slide decks composed of more than 20 slides from slideshare[2] and covering 39 topics. We kept the first 20 slides and truncated the rest of the pages. Then, the workers filtered the collected decks that did not meet the following criteria: (i) the main language is English; (ii) the content is easy for workers to understand; (iii) the decks must contain one or more graphs, tables, figures, or numerical data to avoid creating questions requiring only text-level understanding.

Bounding boxes and categories annotation. To facilitate understanding of the semantic components of images, we annotated all images with bounding boxes and their categories. The workers indicated specific objects in each image by annotating bounding boxes around the objects and classifying them into nine classes that were based on SPaSe (Monica Haurilet and Stiefelhagen 2019) as follows:

- **Title**: presentation title, slide title
- **Page-text**: text in slide, bullet-point text list, text list
- **Obj-text**: text in a figure, image, diagram or table
- **Caption**: description of figure, image, diagram, or table
- **Other-text**: footnote, date, affiliation, code, URL
- **Diagram**: a graphical representation of data, a process
- **Table**: data arranged in rows and columns
- **Image**: drawing, logo, map, screenshot, realistic image
- **Figure**: graph with data points and coordinates

Single-hop QA creation. We asked the workers to create 12,466 QA pairs by selecting a single slide image from a slide deck. The selected slide can be used as evidence to tell whether a system arrived at the right answer for the

[2]https://www.slideshare.net/



(a) Bounding box categories. (b) Reasoning types. (c) Answer types.

Figure 2: Distribution of bounding box categories, reasoning types, and answer types in the test set.

right reasons. We encouraged questions that needed numerical reasoning, including operations of arithmetic expressions with $\{+, -, /, *\}$, counting, and comparisons. Additionally, the workers avoided creating questions that (i) contained selected page numbers; (ii) required external knowledge; (iii) were common to all of the slides (e.g., "What is the title?").

Multi-hop questions creation. We created 2,018 QA pairs for multi-hop reasoning by editing the single-hop questions created in the previous step. For example at the left of Figure 1, "North" is replaced by the phrase "the region with 70% of journals". To this end, we first identified one or two bridge entities in the created questions, and the workers selected related slides as evidence that mentioned the identified ones. Then, the content of the selected slides was utilized to replace the entities in the created questions. The process of creating multi-hop questions by editing may produce unnatural questions, as mentioned in the "Limitations" section, but is easily scalable. A similar approach was taken with MultiModalQA (Talmor et al. 2021), which requires multi-hop reasoning over text, tables, and images in Wikipedia.

Arithmetic expression annotation. We provided arithmetic expressions like "30 - 28" in which the final numerical answer can be arrived at with the four arithmetic operations. The interpretation of the answer generation process is important for creating explainable QA models.

Statistics and Analysis

SlideVQA contains 14,484 QA pairs from 2,619 slide decks, consisting of 52,480 slide images annotated with 890,945 bounding boxes. We split the dataset into 10,617 questions for training, 1,652 (2,215) questions for development (test), making sure that each deck appears in the same split. We compare SlideVQA with related datasets in terms of "Images" and "Questions and Answers".

Images. SlideVQA provides the largest number of images covering broad range of topics among the datasets shown in Table 1. Moreover, SlideVQA provides the largest number of bounding box annotations, where the number of the annotations in SlideVQA is 14.7 times that of VisualMRC. Figure 2a shows the distribution of bounding boxes broken down into nine categories, which cover all classes, including visually related ones (Image and Figure), unlike DocVQA and DocCVQA. To analyze the OCR tokens, we extracted the text shown in the images by using the Google Cloud Vision API[3]. As a result, the number of OCR tokens the system should consider simultaneously is larger (1488.88 tokens) than those of single-image document VQA datasets; the largest dataset (InfographicVQA) has 217.89 tokens.

Questions and answers. As shown in Table 1, SlideVQA requires complex reasoning including single/multi-hop, and numerical reasoning. Figure 2b shows the diverse distribution of questions related to reasoning types. 49.3% of the questions require multi-hop or numerical reasoning. Moreover, SlideVQA is the first dataset to provide annotations of arithmetic expressions for improving numerical reasoning. Figure 2c shows that multi-span and non-span account for 32.4% of the answers, indicating systems also need to generate answers as well as extract multiple spans.

Our Model

Figure 3 shows an overview of our model, called M3D (**M**ulti-**M**odal **M**ulti-image **D**ocument VQA model). We use Fusion-in-Decoder (FiD) (Izacard and Grave 2021), which is a state-of-the-art multi-text encoder-decoder model, as our base model and initialize FiD with a pre-trained T5 (Raffel et al. 2020). We extend FiD to perform the end-to-end SlideVQA task (defined in MAINTASK) by (i) performing evidence selection and question answering tasks as a unified sequence-to-sequence format using multi-task learning, (ii) predicting arithmetic expressions as intermediate reasoning steps instead of generating answers directly to enhance numerical reasoning, and (iii) modifying the input sequence to learn the visual layout and content of the image.

Multi-modal Task-Specific Input

Input token sequence. For each image I_k, we first use Faster-RCNN (Ren et al. 2015), which was trained on SlideVQA, to extract N semantic regions (bounding boxes) and

[3]https://cloud.google.com/vision

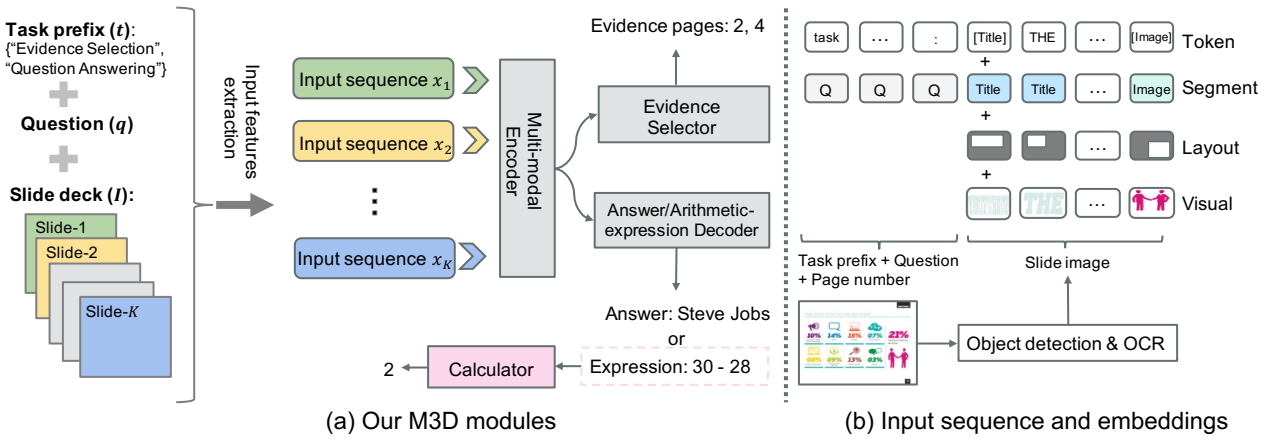


Figure 3: (a) Our encoder-decoder model architecture and (b) input representations. Given a question with a task prefix and a slide deck, the model outputs a corresponding answer/arithmetic-expression and evidence pages. The calculator outputs the final answer to calculate the generated arithmetic expression.

their labels (e.g., Title and Image). We parse the slide image for each extracted region r by using an OCR engine and apply a sub-word tokenizer to obtain OCR tokens $\mathbf{W}_k^r = \{w_{k,1}^r, \ldots, w_{k,n}^r\}$ and corresponding OCR bounding boxes. To jointly train the evidence selection and question answering tasks, we add different task prefixes $t \in \{\texttt{Evidence Selection}, \texttt{Question Answering}\}$ to the encoder input. Specifically, the input sequence is as follows:

$$x_k = (\texttt{task:}t\ \texttt{question:}q\ \texttt{page:}e_k\ \texttt{context:}c_k),$$

where the sequence concatenates each slide and page number pair (c_k, e_k) with the question q and task prefix t. To tell the role of each region, we insert region labels $[\mathrm{R}_k^{r_i}]$, corresponding to the region label of the i-th region r_i in k-th page, before the OCR tokens $\mathbf{W}_k^{r_i}$ extracted in r_i:

$$c_k = ([\mathrm{R}_k^{r_1}], \mathbf{W}_k^{r_1}, [\mathrm{R}_k^{r_2}], \mathbf{W}_k^{r_2}, \ldots, [\mathrm{R}_k^{r_N}], \mathbf{W}_k^{r_N})$$

Input embedding. Following LayoutT5 (Tanaka, Nishida, and Yoshida 2021), the input embeddings $\mathbf{z}$ of the encoder are defined by utilizing multi-modal information, including token $\mathbf{z}^{\text{token}}$, segment $\mathbf{z}^{\text{seg}}$, layout $\mathbf{z}^{\text{lay}}$, and visual embeddings $\mathbf{z}^{\text{vis}}$ as follows:

$$\mathbf{z} = \mathrm{LN}(\mathbf{z}^{\text{token}} + \mathbf{z}^{\text{seg}} + \mathbf{z}^{\text{lay}} + \mathbf{z}^{\text{vis}}) \in \mathbb{R}^{L \times d},$$

where LN is a layer normalization (Ba, Kiros, and Hinton 2016), and L and d are the length of the input sequence and a hidden vector size, respectively. The segment embedding indicates which regions are included in the input sequence. The layout embedding denotes the encoded bounding box coordinates of the token within the image. We normalize all coordinates by the size of images and use embedding layers to embed x-axis and y-axis features separately. The visual embedding is the appearance feature of each region and the OCR bounding boxes, which were obtained from Faster-RCNN. Note that the layout and visual embeddings are set to zero vectors for the task prefix, question, and page number.

Multi-modal Encoder-Decoder

Multi-modal encoder. Our encoder is a stack of m Transformer blocks, consisting of a self-attention layer and a fully-connected layer with residual connections. Following FiD (Izacard and Grave 2021), all K input sequences are encoded independently and then concatenated to form a unified input representation. Formally, we transform each input sequence x_k into $\mathbf{x}_k \in \mathbb{R}^{L \times d}$ and concatenate them into $\mathbf{X} \in \mathbb{R}^{K \times L \times d}$.

Answer/Arithmetic-expression decoder. Our decoder is another stack of m Transformer blocks similar to the multi-modal encoder, where each block has an additional layer of cross-attention between the output sequence and $\mathbf{X}$. The answer decoder is modeled as a conditional generation $p_\theta(y|\mathbf{X})$, where θ represents the set of all model parameters. To allow the model to perform numerical reasoning, we train the system to predict annotated arithmetic expressions y' (e.g., "30 − 28") instead of numeric values y (e.g., "2") by modeling $p_\theta(y'|\mathbf{X})$. During inference, the model itself decides whether numerical reasoning is required or not for each question by predicting an indicator token `Answer:` or `Expression:` at the beginning of the output sequence.

Evidence selector. The selector shares the weights and the architecture of the answer/arithmetic-expression decoder. Instead of only modeling answer generation, we devise a simple method to train evidence selection in a unified sequence. Specifically, we define the output sequence as $\hat{\mathbf{I}}_{\text{pages}} = (\texttt{Evidence pages:}\ \hat{e}_1, \ldots, \hat{e}_{K'})$, where each $\hat{e}$ is the page number of the selected slide.

Training and inference. Our model is trained by minimizing the weighted sum of two losses $\mathcal{L} = \mathcal{L}_{\text{dec}} + \mathcal{L}_{\text{sel}}$, where $\mathcal{L}_{\text{dec}}$ and $\mathcal{L}_{\text{sel}}$ are the negative log-likelihood between the ground-truth and the prediction regarding the decoder and selector, respectively. During inference, we obtain the final prediction to post-process the decoded sequence by removing the task indicator. If an arithmetic expression is gen-

erated (i.e., `Expression:` is generated), we use a calculator to obtain the final results.

Experiments

Experimental Setup

We conducted experiments on the SlideVQA task, evidence selection task, and question answering task respectively defined in MAINTASK, SUBTASKS 1 and 2.

Main task baselines. We mainly evaluated pipeline models as baselines, consisting of evidence selection that produces top-3 evidences and question answering that takes the selection results as input. Here, we introduced a hierarchical LayoutLMv2 (H-LayoutLMv2) inspired by (Tu et al. 2020; Xu et al. 2021), which encodes all slides simultaneously by using another Transformer layer, as the evidence selector. It achieved 96.0% on Recall@3 on the test set. We used three generative QA models: a textual model **T5** (Raffel et al. 2020), a numerical and multi-hop model **PreasM** (Yoran, Talmor, and Berant 2022), and a document VQA model **LayoutT5** (Tanaka, Nishida, and Yoshida 2021). We also used an extractive document VQA model **LayoutLMv2** to predict the single span.

Evidence selection baselines. We also evaluated the evidence selection task alone. **BM25** (Robertson, Zaragoza et al. 2009) is a non-neural retrieval framework to estimate the relevance of texts to a search query. For the neural models, **CLIP** (Radford et al. 2021) encodes the question and each image to predict the highest similar pair. BM25 and CLIP used the top-1 slide as the prediction. **BERT** (Devlin et al. 2019) is a pre-trained language model which only uses text information with the Transformer architecture. **LayoutLM** (Xu et al. 2020) incorporates layout information into the input embeddings of BERT. **LayoutLMv2** includes image features produced by a CNN backbone in input embeddings. To model the interactions between the slides, we used **H-LayoutLMv2** described in the previous section. For neural evidence selection baselines (except for CLIP), we use a hidden state of `[CLS]` in the last layer to feed into an MLP classifier with a sigmoid activation. Evidence is selected if its confidence of binary classification is above the optimal value on the development set.

To evaluate the effectiveness of our generative evidence selection module, we introduced **BinaryClass** as a classification baseline, which uses a two-layer MLP classifier with a sigmoid activation on top of each encoder representation at the start-of-sequence. We also introduced a generative baseline, **ChainGen**, which generates a sequence of selected slide page numbers before the answer (Wei et al. 2022).

Question answering baselines. In addition to the pipeline models, we developed **Q-only**, which takes only the question into T5. We also used a VideoQA model **UniVL** (Luo et al. 2020) that can take all of the slide images as input. Furthermore, we evaluated our base model **FiD**.

Human performance. We asked six crowdworkers (not among those recruited to collect our dataset) to select slide images relevant to the question and answer the question.

Evaluation metrics. Following HotpotQA (Yang et al. 2018), we used exact match (EM) and F1 on each question answering and evidence selection task and also used Joint EM (JEM) and Joint F1 (JF1) to evaluate both tasks. These joint metrics penalize models that perform poorly on either task and assess the accuracy and explainability of the question answering models.

Implementation Details

We implemented all of the models in PyTorch and experimented on eight Tesla V100 32GB GPUs. The size of CLIP was `Large` and the size of the other models was `Base`. We fine-tuned the models using AdamW (Loshchilov and Hutter 2017) with a learning rate of 5e-5 and a dropout rate of 10%, and we linearly warmed up the learning rate over 1000 steps. The batch size was set to 32. We evaluated models every 500 steps and selected the best one on the development set on the basis of the loss. We used a maximum length of 200 tokens for each input sequence of M3D, and set the maximum target sequence length to 50. We trained Faster-RCNN (Ren et al. 2015) with a ResNet-101 (He et al. 2016) backbone by using stochastic gradient descent (SGD) (Ruder 2016) with a learning rate of 1e-3 and batch size of one. Standard anchor scales of [8, 16, 32] and anchor ratios of [0.5, 1.0, 2.0] were used. For the VideoQA baseline, we created a new video at a rate of five frames per second. We used the Google Cloud Vision API to extract text and bounding boxes from images. When the OCR word is tokenized into sub-word tokens, the bounding box coordinates of a sub-word token are the same as those of its whole word.

Experimental Results and Analysis

Does our model outperform the baselines? Table 2 summarizes the results of the main tasks. As shown in Table 2a, M3D outperformed the baselines on joint EM/F1, where the metrics evaluate the consistency between the predicted evidence and answers. For the evidence selection task, Table 2b shows that H-LayoutLMv2 and M3D performed better than the baselines. This indicates that modeling the interaction between multiple slides simultaneously is needed to improve performance. For the QA task, Table 2c shows that M3D outperformed the pipeline methods in all metrics. Our end-to-end M3D model is better at ignoring the slides irrelevant to the question than the answer generator in the pipeline methods that strongly depend on the slides narrowed down by the evidence selector. However, M3D_{GT} in Table 2a achieved a significant improvement by knowing the ground-truth slides. There is room for improving the correctness of evidence selection.

What are the characteristics of our dataset? Table 2 shows that adding modality information tended to improve performance in all tasks. This demonstrates that SlideVQA requires methods to have the ability to jointly understand the text, layout, and visual modalities of documents. As shown in Table 2c, Q-only had the lowest performance, showing that the systems could not answer the question without reading documents in the SlideVQA task. Additionally, UniVL has a comparative result to Q-only, indicating that

Model	Modal	JEM	JF1
PreasM	T	23.4	34.7
T5	T	22.6	34.2
T5 + $\mathbf{z}^{\text{lay}}$	TL	23.6	35.7
LayoutT5	TLV	24.3	36.1
LayoutLMv2†	TLV	16.5	26.5
M3D	TLV	**28.0**	**37.3**
M3D_{GT}	TLV	35.4	44.7
Human	–	88.6	91.9

(a) Performance of main task.

Model	Modal	EM	F1
BM25	T	35.9	47.5
$\text{CLIP}_{\text{zero}}$	V	30.6	34.4
CLIP	V	39.3	43.5
BERT	T	50.3	69.2
BERT + $\mathbf{z}^{\text{lay}}$	TL	52.7	71.0
LayoutLM	TL	42.0	59.9
LayoutLMv2	TLV	51.7	71.5
H-LayoutLMv2	TLV	69.8	**85.6**
M3D	TLV	**75.0**	83.8
Human	–	97.7	98.0

(b) Performance of evidence selection task.

Model	Modal	EM	F1
Q-only	–	10.7	13.5
UniVL	V	10.6	14.1
PreasM	T	30.7	38.2
T5	T	29.3	37.9
T5 + $\mathbf{z}^{\text{lay}}$	TL	31.0	39.7
LayoutT5	TLV	31.7	39.9
LayoutLMv2†	TLV	21.4	29.3
FiD	T	30.4	38.9
FiD + $\mathbf{z}^{\text{lay}}$	TL	30.6	38.9
M3D	TLV	**33.5**	**41.7**
Human	–	89.8	93.0

(c) Performance of question answering task.

Table 2: Performance of SlideVQA tasks. "T/L/V" denotes the "text/layout/visual" modality of images. †denotes the extractive approach. The pipeline models answer the question based on the top-3 evidences obtained by H-LayoutLMv2. M3D_{GT} knows the ground-truth evidence. + $\mathbf{z}^{\text{lay}}$ denotes addition of the layout embedding to the input embeddings. LayoutLM was not pre-trained in any matching task (e.g., text-image matching). $\text{CLIP}_{\text{zero}}$ denotes CLIP without fine-tuning.

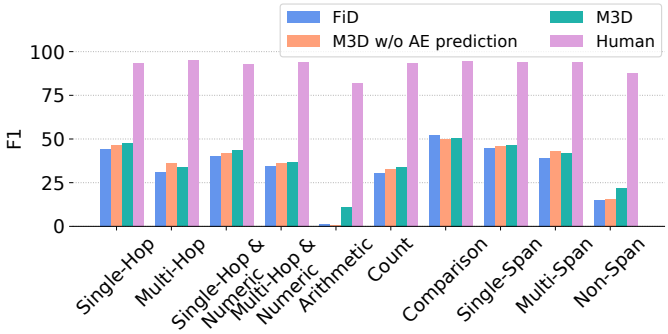


Figure 4: Performance of models and humans on the answer types, reasoning types and numerical operation types in the test set. AE stands for "arithmetic expression".

SlideVQA requires different abilities from VideoQA (Le and Hoi 2020), especially the ability to read texts in images. Tables 2a and 2c show that LayoutT5, a generative model, significantly outperformed LayoutLMv2, an extractive approach. This result is inline with observations on the DROP dataset (Dua et al. 2019), which also has non-span answers (Geva, Gupta, and Berant 2020). Additionally, all of the models performed all of the tasks significantly worse than humans. To be specific, Figure 4 illustrates that (i) better multi-hop reasoning and (ii) non-span answers to questions involving arithmetic operations have to be improved.

Do our sub-modules improve performance? Table 3 lists the results of an ablation study. Here, performance consistently decreased as individual modules were removed from M3D. This indicates that each of the modules is effective. More precisely, the arithmetic expression (AE) generation was influential on the QA and Joint performance, meaning that predicting the arithmetic expression instead of the numerical value enhances the ability to generate answers

Model	Main JEM	Main JF1	Select EM	Select F1	QA EM	QA F1
M3D	**36.2**	**42.8**	**83.1**	**87.7**	**41.3**	**47.1**
w/o AE prediction	35.7	42.3	82.9	87.7	40.5	46.3
w/o Evidence selection	–	–	–	–	40.6	46.4
w/o Layout features	35.1	42.0	82.4	87.1	40.3	46.3
w/o Visual features	34.2	40.9	81.5	86.3	39.0	44.9
w/o Text features	1.0	1.5	8.4	9.8	9.8	12.0

Table 3: Ablation study of M3D on dev set.

Model	Main JEM	Main JF1	Select EM	Select F1	QA EM	QA F1
M3D backbone	–	–	–	–	39.0	44.8
+ BinaryClass	24.7	34.8	54.5	68.5	38.8	44.8
+ ChainGen	34.0	40.8	81.1	86.1	39.8	45.4
+ MultiGen (**Ours**)	**35.7**	**42.3**	**82.9**	**87.7**	**40.5**	**46.3**

Table 4: Performance comparison of different evidence selection methods on dev set.

with numerical reasoning. As shown in Figure 4, applying AE prediction increased F1 by a large margin (+10.4%) in the arithmetic type.

What are the effective evidence selection methods? Table 4 shows that our method, which generates the evidence selection and question answering results separately, obtained the highest performance. It seems that the generative methods (MultiGen and ChainGen) benefited from the text-to-text pre-training of T5 more than the classification-based method (BinaryClass). Our MultiGen decoder that separately trains evidence selection and question answering had the advantage of being easier to train than the ChainGen baseline decoder that trains the two tasks as a single se-

Class	Dev AP	Test AP
Title	86.8	87.5
Page-text	76.9	76.9
Obj-text	29.5	33.4
Caption	25.6	24.9
Other-text	40.5	39.4
Image	60.4	62.2
Diagram	65.4	64.0
Figure	74.1	68.8
Table	67.0	65.6

Table 5: Object detection performance of Faster-RCNN broken down by bounding box categories. We set an intersection-over union (IoU) threshold to 0.5.

	Main		Select		QA	
OCR engine	JEM	JF1	EM	F1	EM	F1
Vision API	**36.2**	**42.8**	**83.1**	**87.7**	**41.3**	**47.1**
Tesseract	22.5	28.3	69.6	74.7	28.3	34.0

Table 6: M3D performance comparison with different OCR engines in the dev set.

quence generation task.

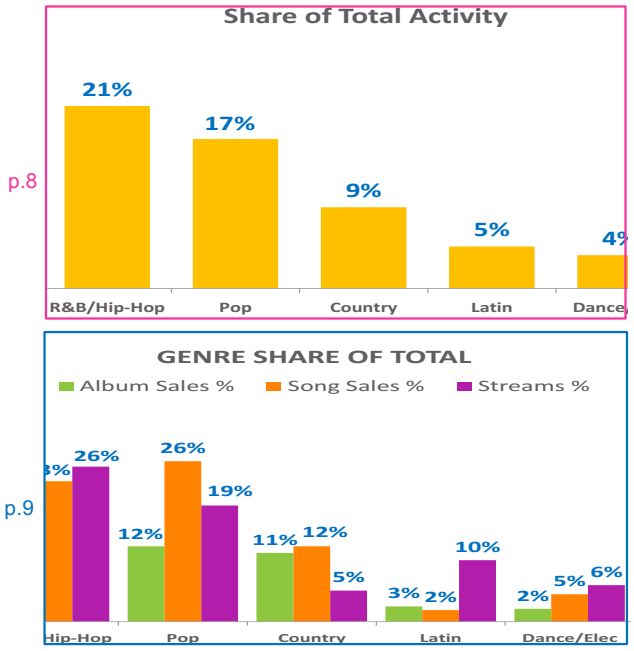
On which categories does the object detection model not work well? Table 5 lists the object detection performance of Faster-RCNN broken down by bounding box categories. These results show that detecting randomly placed and small boxes, such as Obj-text, is more difficult than mostly fixed and large boxes, such as Title.

Does the model performance depend on the OCR engine? Table 6 presents the results of M3D for the Vision API and Tesseract OCR engine. The differences in score are huge for all tasks and show a clear advantage for Vision API. The future direction includes that we will create models showing the robustness to variations in OCR quality.

Qualitative examples. Figure 5 demonstrates our model's performance by visualizing a qualitative example. This example needs multi-hop reasoning and an answer involving an arithmetic operation. FiD gave an incorrect answer because it did not consider the visual layout of the slides. Moreover, while LayoutT5 could not understand the process of getting numerical answers, M3D successfully extracted information ("11%" and "12%") and generated the same answer as the ground-truth.

Discussion and Limitations

SlideVQA is the largest document VQA benchmark that uses multiple images as input and requires multi-hop reasoning; its limitation is that the multi-hop questions created by editing are different from the questions humans might actually ask the system. We argue that developing models that can reason over multiple images is an important research direction, and therefore, we employed an editing method that guarantees multi-hop questions and easily extends the



Q: What is the combined percentage of Album Sales % and Song Sales % for the genre with a 9% Share of Total Activity?

GT	**A**: 23%	**Evidence pages**: 8, 9
FiD	**A**: 57%	**Evidence pages**: None
LayoutT5	**A**: 68%	**Evidence pages**: 8, 9
M3D	**A**: 23% (11% + 12%)	**Evidence pages**: 8, 9

Figure 5: Qualitative example. GT denotes the ground-truth. (·) means the generated arithmetic expression. The slide deck can be viewed at https://www.slideshare.net/musicbizassoc/nielsen-2015-music-biz-presentation-final.

dataset size. Also, our model uses cross-attention on all evidence candidates, which may cause a computational problem when there are a lot of input images (e.g., as in the open-domain QA setting like DocCVQA). To remedy this problem, we consider that models that train a two-stage selector that roughly narrows down candidates to a small number of images and then accurately selects evidence images and an answer generator in an end-to-end manner are promising (Sachan et al. 2021a,b).

Conclusion

We introduced a new document VQA dataset, SlideVQA, focused on the task of understanding slide decks composed of multiple images. We also introduced a unified end-to-end model, M3D, that can perform evidence selection and question answering tasks and enhance numerical reasoning by generating arithmetic expressions. While our evaluation highlighted the promise of this approach, it also revealed a huge gap compared with human performance and several challenges emerge from multi-hop reasoning on multiple images and generating answers with arithmetic operations. We believe that our dataset will contribute to the development of intelligent assistant agents that can comprehend diverse real-world documents.

References

Ba, L. J.; Kiros, R.; and Hinton, G. E. 2016. Layer Normalization. *arXiv:1607.06450*.

Bansal, A.; Zhang, Y.; and Chellappa, R. 2020. Visual question answering on image sets. In *ECCV*, 51–67.

Chen, X.; Zhao, Z.; Chen, L.; Ji, J.; Zhang, D.; Luo, A.; Xiong, Y.; and Yu, K. 2021. WebSRC: A Dataset for Web-Based Structural Reading Comprehension. In *EMNLP*, 4173–4185.

Cui, L.; Huang, S.; Wei, F.; Tan, C.; Duan, C.; and Zhou, M. 2017. SuperAgent: A Customer Service Chatbot for E-commerce Websites. In *ACL*, 97–102.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 4171–4186.

Dua, D.; Wang, Y.; Dasigi, P.; Stanovsky, G.; Singh, S.; and Gardner, M. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *ACL*, 2368–2378.

Fu, T.; Wang, W. Y.; McDuff, D.; and Song, Y. 2022. DOC2PPT: Automatic Presentation Slides Generation from Scientific Documents. In *AAAI*, 634–642.

Geva, M.; Gupta, A.; and Berant, J. 2020. Injecting Numerical Reasoning Skills into Language Models. In *ACL*, 946–958.

Haurilet, M.; Roitberg, A.; Martinez, M.; and Stiefelhagen, R. 2019. Wise—slide segmentation in the wild. In *ICDAR*, 343–348.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Hong, Y.; Wang, J.; Jia, Y.; Zhang, W.; and Wang, X. 2019. Academic Reader: An Interactive Question Answering System on Academic Literatures. In *AAAI*, 9855–9856.

Izacard, G.; and Grave, E. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *EACL*, 874–880.

Kembhavi, A.; Seo, M. J.; Schwenk, D.; Choi, J.; Farhadi, A.; and Hajishirzi, H. 2017. Are You Smarter Than a Sixth Grader? Textbook Question Answering for Multimodal Machine Comprehension. In *CVPR*, 5376–5384.

Le, H.; and Hoi, S. C. H. 2020. Video-Grounded Dialogues with Pretrained Generation Language Models. In *ACL*, 5842–5848.

Lei, J.; Yu, L.; Bansal, M.; and Berg, T. 2018. TVQA: Localized, Compositional Video Question Answering. In *EMNLP*, 1369–1379.

Lei, J.; Yu, L.; Berg, T.; and Bansal, M. 2020. TVQA+: Spatio-Temporal Grounding for Video Question Answering. In *ACL*, 8211–8225.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv:1711.05101*.

Luo, H.; Ji, L.; Shi, B.; Huang, H.; Duan, N.; Li, T.; Li, J.; Bharti, T.; and Zhou, M. 2020. UniVL: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation. *arXiv:2002.06353*.

Mathew, M.; Bagal, V.; Tito, R.; Karatzas, D.; Valveny, E.; and Jawahar, C. 2022. InfographicVQA. In *WACV*, 1697–1706.

Mathew, M.; Karatzas, D.; and Jawahar, C. V. 2021. DocVQA: A Dataset for VQA on Document Images. In *WACV*, 2200–2209.

Monica Haurilet, Z. A.-H.; and Stiefelhagen, R. 2019. SPaSe - Multi-Label Page Segmentation for Presentation Slides. In *WACV*, 726–734.

Powalski, R.; Borchmann, Ł.; Jurkiewicz, D.; Dwojak, T.; Pietruszka, M.; and Pałka, G. 2021. Going full-tilt boogie on document understanding with text-image-layout transformer. In *ICDAR*, 732–747.

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 *ICML*, 8748–8763.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 21(140): 1–67.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *ACL*, 784–789.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *EMNLP*, 2383–2392.

Ren; Shaoqing; He; Kaiming; Girshicka; Ross; Sun; and Jian. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 91–99.

Robertson, S.; Zaragoza, H.; et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.

Ruder, S. 2016. An overview of gradient descent optimization algorithms. *arXiv:1609.04747*.

Sachan, D.; Patwary, M.; Shoeybi, M.; Kant, N.; Ping, W.; Hamilton, W. L.; and Catanzaro, B. 2021a. End-to-End Training of Neural Retrievers for Open-Domain Question Answering. In *ACL*, 6648–6662.

Sachan, D. S.; Reddy, S.; Hamilton, W. L.; Dyer, C.; and Yogatama, D. 2021b. End-to-End Training of Multi-Document Reader and Retriever for Open-Domain Question Answering. In *NeurIPS*, 25968–25981.

Sun, E.; Hou, Y.; Wang, D.; Zhang, Y.; and Wang, N. X. R. 2021. D2S: Document-to-Slide Generation Via Query-Based Text Summarization. In *NAACL-HLT*, 1405–1418.

Talmor, A.; and Berant, J. 2018. The Web as a Knowledge-Base for Answering Complex Questions. In *NAACL-HLT*, 641–651.

Talmor, A.; Yoran, O.; Catav, A.; Lahav, D.; Wang, Y.; Asai, A.; Ilharco, G.; Hajishirzi, H.; and Berant, J. 2021. MultiModalQA: complex question answering over text, tables and images. In *ICLR*.

Tanaka, R.; Nishida, K.; and Yoshida, S. 2021. VisualMRC: Machine Reading Comprehension on Document Images. In *AAAI*, 13878–13888.

Tapaswi, M.; Zhu, Y.; Stiefelhagen, R.; Torralba, A.; Urtasun, R.; and Fidler, S. 2016. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 4631–4640.

Tito, R.; Karatzas, D.; and Valveny, E. 2021. Document Collection Visual Question Answering. In *ICADR*, 778–792.

Tu, M.; Huang, K.; Wang, G.; Huang, J.; He, X.; and Zhou, B. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *AAAI*, 9073–9080.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*, 6000–6010.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E.; Le, Q.; and Zhou, D. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv:2201.11903*.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *KDD*, 1192–1200.

Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florêncio, D. A. F.; Zhang, C.; Che, W.; Zhang, M.; and Zhou, L. 2021. LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. In *ACL/IJCNLP*, 2579–2591.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *EMNLP*, 2369–2380.

Yoran, O.; Talmor, A.; and Berant, J. 2022. Turning Tables: Generating Examples from Semi-structured Tables for Endowing Language Models with Reasoning Skills. In *ACL*, 6016–6031.

Zhang, Q.; Wang, L.; Yu, S.; Wang, S.; Wang, Y.; Jiang, J.; and Lim, E.-P. 2021. NOAHQA: Numerical Reasoning with Interpretable Graph Question Answering Dataset. In *Findings of EMNLP*, 4147–4161.

Zhang, X.; Ramachandran, D.; Tenney, I.; Elazar, Y.; and Roth, D. 2020. Do Language Embeddings capture Scales? In *BlackboxNLP Workshop*.