Heterogeneous Graph Network for MP-DocVQA

Hanwen Zheng
CS Virginia Tech
zoez@vt.edu

Ting-Chih Chen
CS Virginia Tech
tingchih@vt.edu

1. Introduction

In the real world, automatically managing document workflows often involves processing multi-page documents rather than single-page documents, as seen in sectors such as banking, insurance, and public administration. Recently, Document Image Analysis and Recognition (DIAR) methods have demonstrated excellent performance in information extraction and conversion tasks [6–8]. These DIAR methods primarily process information from typed or handwritten text, layout, and graphical elements in single-page documents using Optical Character Recognition (OCR). Differing from single-page documents, multi-page Documents contain significantly more images, graphs, and data segments. Correspondingly, the models ought to infer a more complex hierarchical structure. This task presents three main challenges for Multi-page Document Visual Question Answering (MP-DocVQA). Model comprehension is the first issue. In the previous work, the answer is limited to a single-page document. However, in the MP-DocVQA, the model is more likely to come across the same question based on the different graphs; the important factor is whether the generated answer is derived from the actual required information or from prior learned bias. The second issue is OCR errors. The models are able to correctly ground the questions on the relevant information in the image, but the answers are incorrect due to OCR parsing errors. The input sequence length is the final issue. The size of input tokens from the multi-page document will be constrained in the traditional encoder-decoder transformer models. To address these problems, we will focus on MP-DocVQA dataset [12] and design a novel model based on the knowledge graph.

2. Literature Review

2.1. Document Visual Question Answering

DocVQA challenge [8, 13] focuses on a specific type of visual question-answering task. The models should not only understand the information from the textual content, non-textual elements, layout, and style on a document image with OCR but also respond to the question. DocVQA challenge starts from CVPR2020 and the existing main tasks are single-page Document VQA (SingleDocVQA) [7], document collection VQA [11], infographics VQA [6], and multi-page Document VQA [12]. The single-page Document VQA is a typical VQA style task. The models should explain document images and generate the answer. Also, this is not an n-way classification task. The models must generate text to answer the question, which greatly increases the difficulty. In the document collection VQA, it looks like a retrieval task, where given a question, the results should identify and retrieve all the documents and answer the question from all the documents. The infographics VQA is very similar to the SingleDocVQA task. The difference is that the models mainly extract information from charts and diagrams. In addition, the models should have different methods to process any charts [5].

MATCHA [4], Pix2Struct [3] are image-encoder-text-decoder models using purely visual representations for the SingleDocVQA task. Pix2Struct [3] is a pre-trained with self-supervised pairs of images and target text from web pages to learn parsing data into structural HTML reports. MATCHA [4] is a pre-trained from a Pix2Struct [3] with additional chart understanding and math reasoning knowledge. LayoutLMv3 [2] improves upon LayoutLMv2 [15] by aligning word and patch embeddings in Visual Transformers instead of relying on the output feature map from a pre-trained CNN-based visual encoder. Additionally, it employs unified text and image masking pre-training objectives to better learn multimodal representations. ERNIE-Layout [9] first adopt a serialization model and a reading order prediction task in the textual features encoding process, but still rely on a Faster-RCNN (Ren et al., 2015) as the visual encoder backbone.

CALM [1] and mmLayout [14] utilized a graph-based document representation method in the SingleDocVQA task: CALM [1] extracts purified representations of document contents and keywords from queries with Document

and Question Purifiers. External knowledge from ConceptNet is derived into building knowledge graphs to enhance the common-sense reasoning capability of CALM. mmLayout [14] used both fine-grained and coarse-grained text, vision, and layout information. The model gains structural information through the salient visual region clusters and common sense reasoning through an enhancement strategy that matches text segments with a list of candidate common senses to exploit the semantic information.

In this project, we focus on MP-DocVQA. The object of this task is to answer questions on multi-page document images. Due to the massive scale of the data, we didn't incorporate any external knowledge integration into our work. We believe multi-page level documents have sufficient knowledge within, therefore we hope to utilize document understanding and visual information to extract reasoning paths over a graph-based network. Each question could include 20 pages and the models should clearly point out where the answer is. Since this is a competition, the host does not release the ground truth in the testing dataset. So, we will train on the training dataset and test on the validation set.

2.2. Knowledge Graph

The knowledge graph is a data structure based on the graph. The knowledge graph can connect all different heterogeneous information. This data structure can help the models analyze the relationship between named entities in a knowledge graph. In recent years, with the emergence of artificial intelligence, knowledge graphs have been widely used in chatbots and question-answering systems to assist in the in-depth understanding of human language, support reasoning, and improve the user experience of human-machine interactions, examples include Watson, Siri, Google Allo, and Amazon Echo. In addition, knowledge graphs offer multiple benefits in AI tasks. They can integrate isolated data sources, both structured and unstructured, and extract relationships and insights from hierarchical data while visualizing the flow of information.

The graph-based models are also widely used for document-level information extraction tasks in natural language processing (NLP) and molecule construction in the biomedical field. In Document-level Relation Extraction with Dual-tier Heterogeneous Graph [16], the sentence representations generated by the sentence-to-relation (S2R) attention are refined and synthesized by a heterogeneous graphical convolution network before being fed to the relation-to-sentence (R2S) attention.

To use knowledge graphs in the MP-DocVQA, our main method is to extract information from the document images. Due to the data resource being image type, we will extract image features and text information with OCR. So, our nodes in the knowledge graphs are image type and text type and we will link the nodes with the same entity. This method can efficiently make models to realize the relationship in the multi-page document.

3. MP-DocVQA dataset

The MP-DocVQA (full) dataset inherits all 6,071 documents from SingleDocVQA, totaling over 64,057 pages. The tested MP-DocVQA dataset is filtered to contain 46,176 questions posed over 60,884 page images from 5,928 documents. Each document contains at most 20 pages, and the recognized OCR words reach up to 42,313. 85.95 percent of the questions in the dataset require multiple pages in the document to answer, necessitating a model with strong cross-page information retrieval ability.

The dataset provides OCR annotations extracted with Amazon Textract for all 64,057 document images. During data pre-processing, we extract image-text pairs and bounding boxes from the Amazon Textract results to construct the knowledge graphs.

4. Approach

One difficulty of the task is that documents contain multiple pages and multiple questions; each page can have several questions. In the pre-defined task, each question is processed with the corresponding document, meaning the model will need to read each document multiple times. The MP-DocVQA dataset contains 46,176 questions posed over 60,884 page images from 5,928 documents. Training and testing undoubtedly require a significant amount of computational resources. We propose a method that transforms each document into a heterogeneous graph and performs the VQA task by reasoning on the graph.

4.1. Heterogeneous Graph Network Construction

DocVQA typically requires both visual and textual information to provide answers. Given that a multi-page (MP) document contains a vast number of words, extracting every word entity to construct a knowledge graph is not realistic. Furthermore, for the MP-DocVQA task, the model doesn't need a fine-grained understanding of each entity, but the ability to reason the whole document. To address this issue, our approach is to design a knowledge graph where each node represents the information of each OCR line piece. Therefore, we focus on capturing more

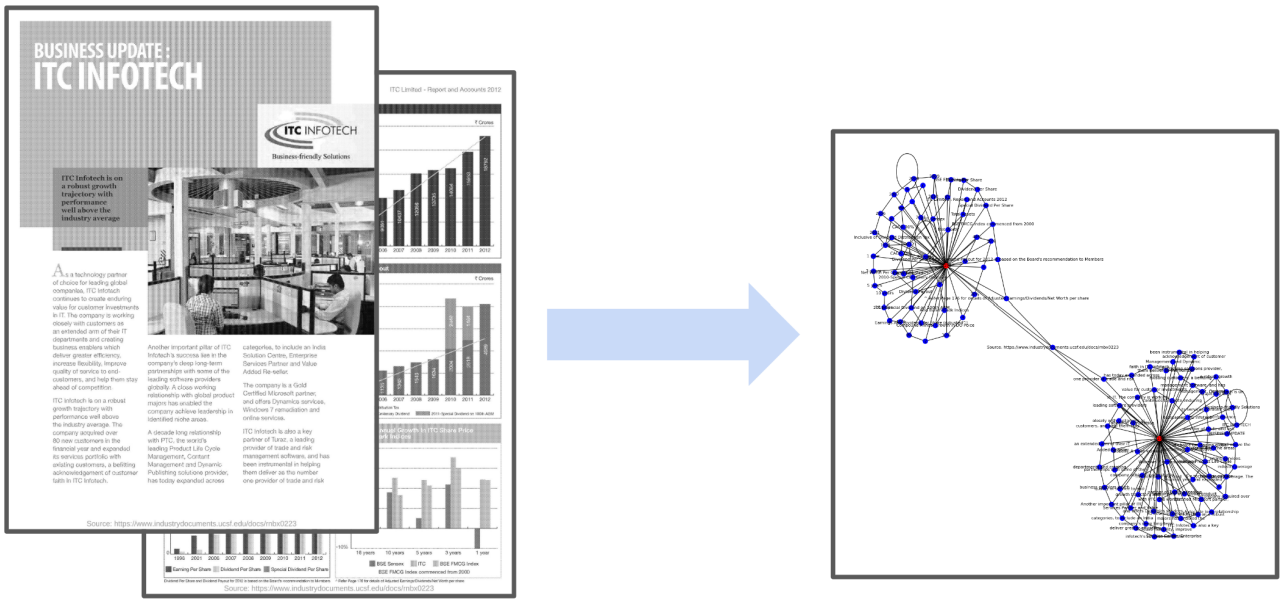

Figure 1. One of the examples from the testing dataset. We build up a knowledge graph with the information from two pages. The red nodes are page id and the blue nodes are information extracted from OCR.

global information in a multi-page setting. We design two heterogeneous graphs G. One is text-only KG and the other is multi-modalities KG as shown in Figure 2. In text-only KG, each G includes question node Q, page nodes P, and text nodes T. Text nodes are the instances directly obtained from the OCR line text. Page nodes are a set of dummy nodes representing the page number. Additionally, in the multi-modalities KG, each G also includes question node Q, page nodes P, and multi-modalities nodes M. Multi-modalities information includes OCR bounding box, word texts, and document images. Figure 1 is the example to construct the knowledge graph.

Edge - Question node to Page nodes: Question node Q is designed to link with each page nodes P by $Q - P$ edges. The reason for this edge is to let the model retrieve the answer page number.

Edge - Page node to Page node: Due to the task being a multi-page DocVQA, we follow the reading order to link with each page node P by $P - P$ edges.

Edge - Page node to Textual nodes: All text entity nodes are fully connected to the indicator page node by $P - T$ edges.

Edge - Page node to Multi-modalities nodes: All multi-modalities information nodes are fully connected to the indicator page node by $P - M$ edges.

Note: All edges initially have no weights. However, following the application of the Graph Attention Network (GAT) layers, these edges acquire representations based on the attention weights between nodes.

4.2. Node Encoding

For node encoding, we have two methods. As a preliminary, for a given single-page document image I, OCR can be applied to the image to get a set of content pieces denoted as $C = \{c^0, c^1, \cdots, c^n\}$ as document tokens, which is typically organized in the natural reading order. In the text-only KG, we use pre-trained language models (PLM) T5-small [10] to encode Q, and T nodes with c^i_{text}; the embedding is the output from the last hidden layer of the T5 Encoder Model. Then in the multi-modalities KG, we encode Q node with T5 and encode M nodes with LayoutLMv3 [2] for obtaining the corresponding textual, visual, and positional embedding of c^i, denoted c^i_{text}, c^i_{image}, and c^i_{box}, respectively. For each piece of content data, the OCR tool offers its 2D location along with the dimensions (width and height) of the bounding box, we will use the coordinate as the positional information of document contextual tokens. For each multi-modalities node, the vector representation M_{embed} of M node is constructed by a multi-modal transformer LayoutLMv3 model, denoted as F_m.

$$M_{embed} = F_m(c^i_{text} + c^i_{image} + c^i_{box})$$

4.3. MP-GNN Model

Our model's architecture involves passing the embeddings from the T5 encoder layer to a Graph Neural Network (GNN) layer, followed by utilizing the GNN layer's output as input for a T5 decoder class, which forms the final layer.

Within the GNN layer, we first implement a densely connected Graph Convolutional Network (GCN) layer on all the T or M nodes. This step updates all content nodes' information by aggregating features from neighboring nodes. Next, we concatenate all updated content nodes to refine the P node representations via a max-pooling message-passing strategy.

Finally, Graph Attention Network (GAT) layers with multi-headed attention are used to acquire the graph's final representation. Considering that different layers of the GNN express features of varying abstract levels, we use the hidden states of the Q node as the final output of the GNN layer to encompass features across all levels.

4.4. Loss function

We employ the cross-entropy loss as the training objective for the MP-GNN generation model. We take the generated output logits from the T5 decoder model, which is the predicted probability distribution of tokens, and also can be seen as the answer token confidence. The GNN layer outputs an attention tensor between the question node and all page nodes. We obtain the attention distribution of all $Q - P$ edges by getting the max argument of the tensor.

The cross-entropy loss measures the dissimilarity between the logits and true distribution of the target sequences as well as between the attention distribution and a one-hot encoding of the correct page index embedding.

$$Loss = Loss_{decode} + \eta Loss_{page}$$

The cross-entropy loss function is widely used for training neural network models in sequence generation tasks due to its effectiveness in capturing dependencies between tokens in the target sequences. During training, the model learns to minimize the cross-entropy loss, resulting in a more accurate and generated answer and page index.

5. Evaluation

In the evaluation, we mainly use the metrics from the original paper [12]. The metrics include the accuracy of the answer page etc. The metric used in our evaluation is the accuracy of the answer page. This metric measures whether the models can accurately predict the correct page index that contains the answer. Although the original paper did not mandate the use of this metric, we believe it can assist us in analyzing the models' ability to extract basic information.

MP-DocVQA defines three baseline task set-ups:

1. **Oracle**, this setup aims at mimicking the SingleDocVQA task, therefore, only the page that contains the answer is given as input. We can create a single-page KG to tackle this setup.

2. **Concat**, the input is the concatenation of the contexts of all the pages of the document. We can perform this setup by removing page nodes and $P - P$ edges in our HG.

3. **Max conf**, each page is processed separately by the model, providing an answer for every page along with a confidence score in the form of logits. Then, the answer with the highest confidence is selected as the final answer with the corresponding page as the predicted answer page.

5.1. Results

Due to time constraints, we managed to train our models only on the first 1000 training instances out of 36230, saving these as checkpoint0. During inference, the model failed to generate meaningful answer texts, possibly due to under-training, decoder errors, or model structural issues. Given that the model outputted unclear results, we focused only on the accuracy of the answer page. For the text-only Knowledge Graph (KG), the accuracy of the answer page was 37.93. For the multi-modalities KG, the accuracy of the answer page increased to 53.00. Despite being under-trained, our models still demonstrate some level of performance when compared with some baseline models for answer page predictions 1. However, a key assumption from this project is that the knowledge graph representation method is superior to the sequence-to-sequence model.

As the MP-DocVQA dataset was derived from the SingleDocVQA dataset, data noise was observed, such as questions like "What does the first paragraph discuss?" when the answer page label is not the first page, making it unclear where the answer might be located.

5.2. Ablations

Our model demonstrates a significantly higher answer page accuracy of 15.07 when utilizing multi-modal em-

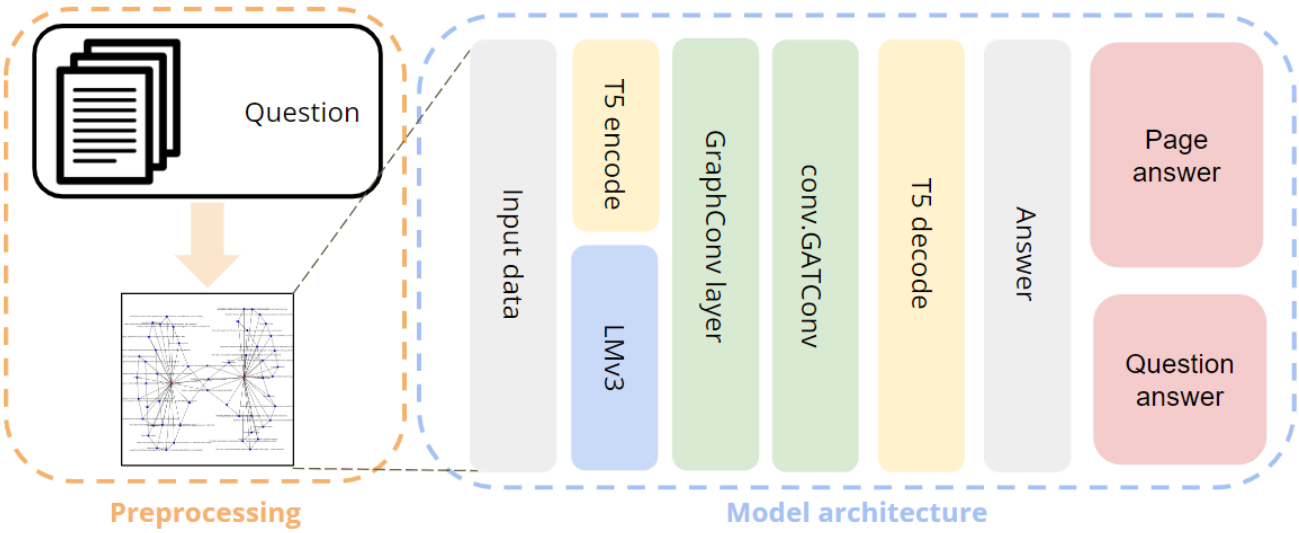

Figure 2. The model pipeline for multi-modalities KG.

Model	Size	Parameters	Max Seq. Length	Setup	Ans. Page Accuracy
BERT	Large	334M	512	Concat	51.61
LayoutLMv3	Base	125M	512	Concat	51.94
Text-only KG	T5-Small	60M	64	Multipage	37.93
Multi-modalities KG	T5-Small	60M	64	Multipage	**53.00**

Table 1. Results

bedding compared to text-only embedding. This shows a substantial improvement compared to the text-only baseline. For future ablation studies, we could conduct more experiments based on the number of GNN layers, such as those with or without a GCN layer, and consider bi-directional GAT layers.

Our MP-GNN Model, encoded with LayoutLMv3, outperforms the pure LayoutLMv3 model by 1.06, verifying the effectiveness of the Knowledge Graph (KG) and GNN. We speculate that enhancing our model with additional GNN layers could further improve its performance.

6. Future Work

We aim to incorporate document information extraction datasets in our future research to further assess our document graph representation. Additionally, we plan to implement a Reading Order Prediction (ROP) strategy to arrange the nodes in an appropriate reading sequence. By deconstructing the document, we can acquire charts, graphs, tables, images, and logos, which will serve as new types of nodes. This could potentially allow us to train our own multimodal embeddings. Furthermore, by assigning weights to edges with co-reference relations, we can improve our document information learning through graph representations.

References

[1] Qinyi Du, Qingqing Wang, Keqian Li, Jidong Tian, Liqiang Xiao, and Yaohui Jin. CALM: Commen-sense knowledge augmentation for document image understanding. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3282–3290. ACM. 1

[2] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. LayoutLMv3: Pre-training for document AI with unified text and image masking. 1, 3

[3] Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. 1

[4] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. MatCha: Enhancing

visual language pretraining with math reasoning and chart derendering. 1

[5] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. 1

[6] Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. 1

[7] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2199–2208. IEEE. 1

[8] Minesh Mathew, Ruben Tito, Dimosthenis Karatzas, R. Manmatha, and C. V. Jawahar. Document visual question answering challenge 2020. 1

[9] Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Teng Hu, Weichong Yin, Yongfeng Chen, Yin Zhang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. 1

[10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. 3

[11] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Document collection visual question answering. 1

[12] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multi-page DocVQA. 1, 4

[13] Rubèn Tito, Minesh Mathew, C. V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. ICDAR 2021 competition on document VisualQuestion answering. 1

[14] Wenjin Wang, Zhengjie Huang, Bin Luo, Qianglong Chen, Qiming Peng, Yinxu Pan, Weichong Yin, Shikun Feng, Yu Sun, Dianhai Yu, and Yin Zhang. mmLayout: Multi-grained MultiModal transformer for document understanding. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4877–4886. ACM. 1, 2

[15] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. 1

[16] Zhenyu Zhang, Bowen Yu, Xiaobo Shu, Tingwen Liu, Hengzhu Tang, Wang Yubin, and Li Guo. Document-level relation extraction with dual-tier heterogeneous graph. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1630–1641, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics. 2