Can seeing documents improve Data efficiency in Document Question Answering?

Venkat Mohit Sornapudi[1,2]

[1] Vrije Universiteit Amsterdam v.m.sornapudi@student.vu.nl
[2] Deloitte vsornapudi@deloitte.nl

Abstract. This thesis investigates the potential benefits of incorporating visual information into Document Question Answering tasks. This is done by comparing the counterparts of the QA and Question-Answer Generation (QAG) models. While Visual QA (VQA) models have shown improved performance compared to Textual QA models on VQA datasets, a fair comparison of these models has not yet been made by evaluating text-only Document QA datasets on which both can excel. So we do it by focusing especially on the data efficiency and domain adaptability of these models. On the other hand, although there exist textual Question-Answer Generation models that can improve Document QA datasets, no visual Question-Answer Generation models are trained explicitly on Document QA datasets. So the study proposes a Visual Question Answer Generation model that can augment Visual QA data in zero-shot and few-shot settings to enhance QA model training and domain adaptability. The results of this study can contribute to the development of more effective models for text-only Document QA tasks even with limited or zero question-answer pairs training data.

Keywords: Document Question Answering · Question Answer Generation · Data efficiency · Multi-modality.

This document explains the project plan of my thesis research.

1 Introduction

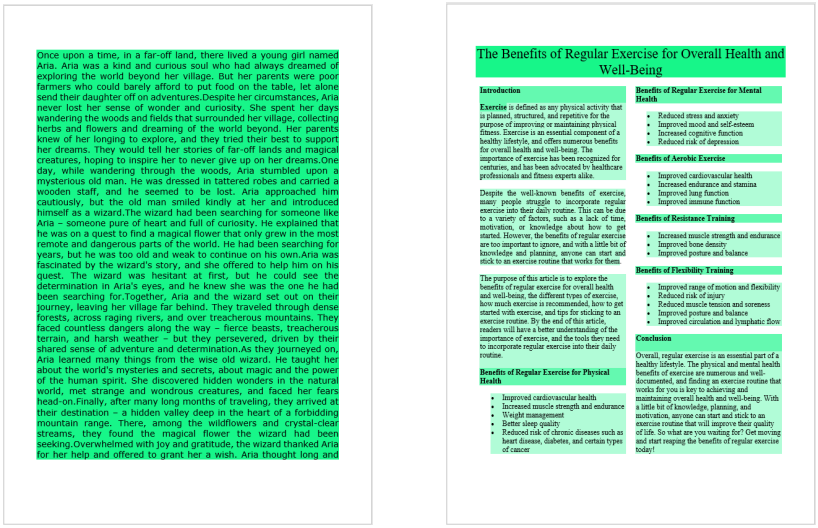
The advent of Artificial Intelligence (AI) has revolutionized the way we approach tedious, time-consuming, and complex tasks. Automation has significantly improved efficiency, accuracy, and decision-making capabilities across various industries. One such task is automatically reading, understanding, and analyzing documents. AI which does this is known as Document AI [2]. With the increasing demand for extracting insights and information from unstructured documents, Document AI has gained significant interest, especially in finance, healthcare, and legal sectors that rely heavily on document processing. As a result, there has been a surge in the development of Document AI products, including DocQMiner, RegMiner, and RegHub. Among all the Document AI tasks, automated

Question Answering (QA) over documents, stands out as the most useful as it allows users to query any information about a given document or set of documents in simple natural language.

On a whole, documents can be classified into layout-independent and layout-dependent categories. As shown in figure 1a, a layout-independent document does not have a structure and the text is presented as a single block. This is kind of documents are very rare. On the other hand, as shown in figure 1b, layout-dependent documents have a structure and the text is presented as a headings, subheadings, sub-subheadings, lists, and other design elements to draw reader's attention. Document layout helps us in reading and understanding the document; hence most the documents are layout-dependent. Even the most primitive structure i.e. a paragraph carry a meaning and can generally be split into topic, supporting and concluding sentences. That is why when we look at a document, we don't process just the text in it but also the layout of the document. For instance, by reading a newspaper, if we want to manually search the names of banks whose market value is falling, we scan the document page by page to get to a few (finance-related) articles excluding sports, films, and science sections. Having an understanding of the layout and relevancy of topics we reduced our search time. A machine can do the same by evaluating the relevancy of topics from text and being visually aware of the layout. Hence we hypothesize that VQA models are more data efficient than Textual QA models. On the other, if we train a QA model on children's literature and want to search the same list of names of banks from a newspaper, the chance of finding it is lesser than if the QA model is trained on financial documents. This is because the domains of training data are different and the kind of questions change. Even though the QA model on children's literature can answer, most probably the model will think of the "bank" as the edge of a river or stream but not as a financial organization. This underperformance is due to a mismatch of training and testing domains is called the domain adaptability problem.

Document QA using text modality alone is termed as Textual QA. Document QA that combines text and vision modality is termed as Visual QA (VQA). VQA models outperformed their textual counterparts on DocVQA dataset [4], however, it is not determined they show same dominance on any text-only document dataset, in which questions are posed only on text contained in them. One of the objectives of this study is to assess and compare the data efficiency and domain adaptability of architecturally similar VQA models and Textual QA models on text-only VQA datasets. To understand the data efficiency of QA models, these models are evaluated at different levels of question-answers availability. To understand data adaptability, the same models trained on a set of domains are evaluated on different sets of domains.

Data augmentation is a powerful technique that can improve the training of models, especially in low-resource conditions. In Document QA, it can be done by creating new document-question-answer tuples or question-answer pairs. We focus on the latter method. Though there exist both textual [30,31,32,33,34,37] and visual [35,36] Question-Answer Generation models (QAG), they are not best



(a) Layout-independent document (b) Layout-dependent document

Fig. 1: Human annotated layout attention. Here, 3 levels of attention is depicted by the highlighting the text. The darker the highlighting, the higher the attention.

suited for Document QA. The textual models focus only on textual features (not visual). The existing visual models are not trained on textually heavy documents, which documents generally are. So it is not yet been determined whether how well the vision information can help in generating better question-answer pairs. To cover this research gap, a novel modular-based VQAG model will be proposed that can be used as a pre-trained model for question-answer generation. The VQAG (Visual-QAG) model will be tested in zero-shot and few-shot scenarios on a few datasets by feeding it with zero and a few of their question-answer pairs to evaluate its domain adaptation capability. A comprehensive qualitative and quantitative analysis will be done on these QA generations.

Overall, the study aims to investigate the extent to which multi-modality can assist in text-only QA tasks. Due to the limited period of the thesis, the scope of the research is restricted to single-page single-document single-hop/single-span short-answer-type text-only QA tasks that do not require numerical reasoning, table understanding, and understanding objects in images (infographics/photos) such as bar sizes in a bar plot or identifying an object in a person's hand. But, this work can be easily extended to multi-page text-only QA tasks (following [6]) and multi-hop/multi-span text-only QA tasks (without any modifications).

To summarize, the main hypotheses (H) and research questions (RQ) of the thesis are as follows:

1. **Data efficiency and domain adaptability of QA models:**
   RQ: Which QA models are more data efficient and domain adapt-

able for text-only documents?
H: VQA models are more data efficient and domain adaptable than their Textual equivalents when evaluated on text-only documents. In particular, the performance of VisualBERT [24] is better than BERT [23], LLMv3 [20] is better than RoBERTa [21], LayoutXLM [22] is better than XLM [25] on WebSRC [9], and SQuAD 2.0 [16] datasets.

2. **Question-Answer Generation:**
RQ: Is it possible to develop a Visual QAG model that does well in both zero-shot and few-shot settings?
H: Based on Textual QAG models, a pre-trained VQAG model can be developed using SQuAD 2.0 [16] dataset. It will be more domain adaptable and makes QA model training more data efficient compared to state-of-art Textual QAG models on WebSRC [9] dataset.

2 Related work

2.1 Question-Answering

Models

A document can be represented through two modalities: text and vision. This means that there are two types of QA models: Textual QA models that utilize only the text modality, and VQA models that incorporate both text and vision modalities. So far, encoder-based Large-scale Language Models (LLMs) have shown the greatest promise for the QA task, owing to their powerful self-attention and cross-attention capabilities [2]. As a result, this work exclusively studies and employs transformer-based QA models.

Textual QA models: The core of a Textual QA model is a sequence-to-sequence model that take text embeddings of a given context (text) as well as the given question and spit out the answer embeddings. So general encoder-based LLMs can do QA. Even though bigger and more complex state-of-art models such as PaLM [15], T5 [29] exist, in this work we focus on the general encoder-based LLMs i.e. RoBERTa [21], BERT [23], and XLM [25] because they are the backbone transformers of models that we want to compare with.

VQA models: The core of a VQA model is a sequence-to-sequence model that gives out answer embeddings by using text embeddings of the given question and OCR text, image embeddings of image(s) of the document, and positional embeddings between the text and image(s). Apart from producing these embeddings, depending upon the dataset, the model might even need to do OCR extraction of the text and bounding boxes as a pre-processing step over the document image(s). Several of these models have already been developed, out of which TiLT [27] models is the state-of-the-art, to which ERNIE-Layout [26] gives a close battle. As it does not have a textual equivalent we resort to VisualBERT [24], LLMv3[20], and LayoutXLM [22].

The frequently used evaluations are BLEU (n-gram), METEOR, ROUGE-L, CIDEr, BERTScore, and ANLS.

Datasets All the datasets, which are suitable for this work, along with their characteristics are mentioned in table 1. All the keywords of the table will be explained in the final thesis report. The suitable datasets contain documents as images or HTML code (which can be converted into image). Due to the scoping of the project, only the first block of datasets i.e. WebSRC [9], and SQuAD 2.0 [16] will be used. WebSRC dataset contains question-answer pairs in CSV format and documents as PNG, JSON (OCR text), and HTML formats sorted into different domains. SQuAD 2.0 contains documents-question-answer tuples in JSON format where the documents are wiki articles that can be identified using the article title.

2.2 Question-Answer Generation models

After conducting extensive research, eight relevant question-answer generation models have been found. In the final thesis report, we will provide a comprehensive discussion of these models.

Textual QAG models: [30,31,32,33,34,37]

Visual QAG models: There are very few available [35,36]. These are not trained on documents but rather on real-world photos.

The existing models can be categorized into the following 3 types:
(c = *context*, q = *question*, a = *answer*)

1. Answer-first question-next generation [32,34]: $P(a|c), P(q|c,a)$
2. Question-first answer-next generation [32,30,38,31]: $P(q|c), P(a|c,q)$
3. Simultaneous question-answer generation [32,33,37,36,35]: $P(q,a|c)$

Of all these types of QA generation, question-first answer-next generation is the most promising one as it can increase question diversity and coverage [30]. These models are exactly/loosely based on 3 modules: Answer Entity Recognition (AER), Question Generation (QG), Question-Answering Extractor (QAE). Here the primary input is a document or context. AER module extracts an answer entity from the primary input which is similar to final answer (during training loss is computed as difference between this entity and final answer). QG module uses the extracted answer entity and the primary input to generate a question. Finally, the QAE module is a general QA model that spits out an answer for given question and primary input.

We focus on round-trip consistency [31] as it is the most simple and widely used one (it has been used ever since back-translating was proposed for neural machine translation). It is best suited for question-first answer-next generation where the final output of the pipeline is generated answer. The consistency is checked by comparing the generated answer and answer entity.

The frequently used measures for QA generation evaluations are EM (exact match), F1 scores.

3 Methodology

3.1 Data efficiency and domain adaptability of QA models

To compare the data efficiency of VQA and Textual models, line plots between *test metric* vs *amount of training data used* will be created for BERT [23] vs VisualBERT [24], RoBERTa [21] vs LLMv3 [20], and XLM [25] vs LayoutXLM [22]. As these are generic language or document understanding models that are not pretrained on wikipedia data, WebSRC [9], and SQuAD 2.0 [16] datasets are used for their fine-tuning (training). This will be done by logarithmically increasing the training QA pairs from zero to full training set from respective VQA datasets. Later the trained models will be evaluated over respective VQA test datasets. The test metrics will be averaged over multiple seeds of random selection of these training QA pairs for fair comparisons.

To compare the domain adaptability of VQA and Textual models, first the models are fine-tuned (trained) on a few domains such as science, entertainment of WebSRC [9] and SQuAD 2.0 [16]. Later, a table will be generated comprising their test results over other domains such as finance, sports from the same datasets. But, for this datasets are to be split into different domains. Here, BERT [23] vs VisualBERT [24], RoBERTa [21] vs LLMv3 [20], and XLM [25] vs LayoutXLM [22] comparisons are made.

3.2 VQAG model

Our modular VQAG model is based on the question-first answer-next generation approach. The modules of the model are AER, QG, and QAE (mentioned in related work). The architecture is inspired by [30]. Except AER all other components are derived from [30] because the previous work's AER relies on sentence understanding and cannot accommodate image as additional input. Hence, a new kind of AER is proposed which is inspired by the encoder of [34]. This previous work encodes given context, classifies relevancy of sentences and uses pre-trained LLMs such as UniLM, ProphetNet, and ERNIE-GEN as decoders to generate questions. Similar to [30], a BART [28] model will be used for QG module but with multi-modality. BART is one of the best-suited models because it can understand the context of input from left to right as well as right to left (due to the transformer encoder) and performs well at sequence generation due to its auto-regressive nature (of transformer decoder). QAE module is just a QA model of choice. During training, AER and QG are trained together while using a fine-tuned QA model (trained on pre-training dataset separately) as QAE. No filtration is done during training.

4 Experimental setup

4.1 Data efficiency and domain adaptability of QA models

Research design A quantitative analysis will be done to compare Textual QA and VQA models in terms of their data efficiency and data adaptibility.

Data collection We selected WebSRC [9] and SQuAD2.0 [16] for the purpose of experiments. WebSRC already contains OCR text, the bounding boxes and documents that are separated in to each domain. On the other hand, SQuAD2.0 doesn't all of those. So pytessaract or LLMv3's Microsoft OCR is used for this purpose.

Exploratory data analysis The following information is derived from each dataset:

1. Question, answer, document text lengths histograms.
2. Any overlapping samples in both datasets.
3. Distribution of data over each domain.
4. Histogram of count of question-answer pair for each document.
5. Pie charts of first 3 words of questions.
6. Key statistics: [percentages of distinct questions, training/dev/test questions (answerable and unanswerable)], [average question, answer, document text lengths], [average number of answers per question], [number of distinct words in questions, answers, document text]

Experimental procedure The experimental steps are as follows:

1. Pre-process datasets: OCR and bounding boxes extraction, domain separation, create embeddings, exclude questions on images and tables.
2. Implement Textual QA and VQA models after thorough comparison of their architectures and number of parameters.
3. Generate line plots to understand data efficiency and a table for domain adaptability as mentioned in the methodology.
4. Draw conclusions from them by hypothesis testing and inference times.

Evaluation metrics : BLEU (n-gram) and ANLS

4.2 VQAG model Pre-training

1. Figure out best input representations, AER and QG loss expressions, and metrics for pre-training tests.
2. Build module after module.
3. Hyper-parameter optimization (including loss-balancing parameter) of whole pipeline.

4.3 VQAG model Inference

Inference will be done on WebSRC dataset.

Zero-shot VQAG The pre-trained AER and QG from this work and pre-trained QA model from elsewhere, as QAE, are jointly used to create zero-shot synthetic QA pairs by just passing OCR text, bounding boxes and image of a document. The QA pairs are filtered by round-trip consistency.

Few-shot VQAG In few-shot QA generation setting, AER and QG from this work and pre-trained QA model from elsewhere, as QAE, are jointly trained on few QA pairs (selected at random) from the training set. This would help the VQAG model to adapt to the particular domain. Now, this fine-tuned VQAG model is used to generate synthetic QA pairs. The QA pairs are filtered by round-trip consistency.

1. Answer qualitative questions on few-shot synthetic data:
   What kind of questions (closed/extractive/...) are generated? Are they similar to the original datasets?
2. Answer quantitative questions on few-shot synthetic data:
   How do the distributions of the number of questions generated per document and lengths of questions generated on the datasets look? Are they similar to the original datasets? How is the distribution of questions by their first three words (pie-chart)?

5 Planning

In figure 2, the project timeline is proposed in form of a Gantt chart.

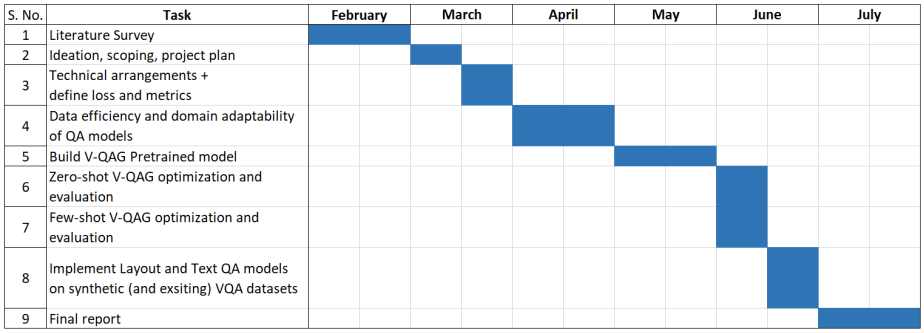


Fig. 2: Planned timeline

6 Appendix

References

1. Henri Bal, Dick Epema, Cees de Laat, Rob van Nieuwpoort, John Romein, Frank Seinstra, Cees Snoek, and Harry Wijshoff: "A Medium-Scale Distributed System for Computer Science Research: Infrastructure for the Long Term", IEEE Computer, Vol. 49, No. 5, pp. 54-63, May 2016.

2. Cui Lei: "Document AI: Benchmarks, Models and Applications", ICDAR, https://www.microsoft.com/en-us/research/publication/document-ai-benchmarks-models-and-applications-presentationicdar-2021/, 2021.
3. Alanazi, Sarah & Mohamed, Nazar & Jarajreh, Mutsam & Algarni, Saad. (2021). Question Answering Systems: A Systematic Literature Review. International Journal of Advanced Computer Science and Applications. 12. 10.14569/IJACSA.2021.0120359.
4. Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021b.
5. Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. Visualmrc: Machine reading comprehension on document images. arXiv preprint arXiv:2101.11272, 2021.
6. Hierarchical multimodal transformers for Multi-Page DocVQA
7. Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. Towards complex doc- ument understanding by discrete reasoning. In Proceedings of the 30th ACM International Conference on Multimedia, pages 4857–4866, 2022
8. Minesh Mathew, Viraj Bagal, Rub'en P erez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V Jawahar. Infographicvqa, 2021a
9. Xingyu Chen, Zihan Zhao, Lu Chen, JiaBao Ji, Danyang Zhang, Ao Luo, Yuxuan Xiong, and Kai Yu. 2021. WebSRC: A Dataset for Web-Based Structural Reading Comprehension. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4173–4185, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
10. DuReadervis: A : A Chinese Dataset for Open-domain Document Visual Question Answering](https://aclanthology.org/2022.findings-acl.105) (Qi et al., Findings 2022)
11. Tsatsaronis, G., Balikas, G., Malakasiotis, P. et al. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. BMC Bioinformatics 16, 138 (2015). https://doi.org/10.1186/s12859-015-0564-6
12. NewsQA: A Machine Comprehension Dataset](https://aclanthology.org/W17-2623) (Trischler et al., RepL4NLP 2017)
13. Natural Questions: a Benchmark for Question Answering Research
14. MultiModalQA: complex question answering over text, tables, and images
15. PaLM: Scaling Language Modeling with Pathways
16. Know What You Don't Know: Unanswerable Questions for SQuAD
17. Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. "A Dataset for Document Visual Question Answering on Multiple Images". In Proc. of AAAI. 2023.
18. Rub'en Tito, Dimosthenis Karatzas, and Ernest Valveny. Document collection visual question answering. In International Conference on Document Analysis and Recognition, pages 778–792. Springer, 2021.
19. https://bit.ly/36O2Vow
20. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking
21. RoBERTa: A Robustly Optimized BERT Pretraining Approach
22. LayoutXLM: Multimodal Pre-training for Multilingual Visually-rich Document Understanding
23. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (https://aclanthology.org/N19-1423) (Devlin et al., NAACL 2019)
24. VisualBERT: A Simple and Performant Baseline for Vision and Language
25. Cross-lingual Language Model Pretraining

26. ERNIE-Layout: Layout Knowledge Enhanced Pre-training for Visually-rich Document Understanding
27. Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer
28. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension
29. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer
30. Cooperative Self-training of Machine Reading Comprehension (https://aclanthology.org/2022.naacl-main.18) (Luo et al., NAACL 2022)
31. Synthetic QA Corpora Generation with Roundtrip Consistency (https://aclanthology.org/P19-1620) (Alberti et al., ACL 2019)
32. End-to-End Synthetic Data Generation for Domain Adaptation of Question Answering Systems (https://aclanthology.org/2020.emnlp-main.439) (Shakeri et al., EMNLP 2020)
33. Simple and Effective Semi-Supervised Question Answering
34. Learning to Generate Questions by Enhancing Text Generation with Sentence Selection
35. Cross-Modal Generative Augmentation for Visual Question Answering
36. TAG: Boosting Text-VQA via Text-aware Visual Question-answer Generation
37. Wang, Tong and Yuan, Xingdi (Eric) and Trischler, Adam
38. Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. arXiv preprint arXiv:2104.08678

Datasets	Lang.	Text-only QA	Multi-hop/ Multi-span	Multi-page	Multi-doc	Needs image understanding	Comments
WebSRC [9]	En	Yes	No	No	No	No	HTML source code, screenshots and metadata; answer is either a text span or yes/no
SQuAD2.0 [16]	En	Yes	No	No	No	No	web-page can be extracted using wikipage title
DocVQA [4]	En	No	No	No	No	No	mix of printed, typewritten and handwritten content; extractive answers
VisualMRC [5]	En	No	No	No	No	No	web-pages; mostly short sentences/phrases; abstractive answers
NewsQA [12]	En	–	Yes	No	No	No	HTML source code; short answers
Natural Questions [13]	En	–	Yes	No	No	No	web-page; short + long answers
DuReader$_{vis}$[10]	Zh	–	Yes	No	No	No	noisy texts; answers contain long answers such as multi-span texts, lists, and tables
Insurance VQA [19]	Zh	–	–	–	–	–	scanned documents of insurance scenarios (for example: medical bills)
MP-DocVQA [6]	En	–	Yes	Yes	No	No	has sections, paragraphs, diagrams, table
TAT-DQA [7]	En	–	Yes	Yes	No	No	+tables; requires numerical reasoning
MultiModalQA [14]	En	–	Yes	No	No	Yes	web-page can be extracted using wikipage url
InfographicVQA [8]	En	–	Yes	No	No	Yes	has infographics; requires numerical reasoning; extractive answers
SlideVQA [17]	En	–	Yes	Yes	No	Yes	requires numerical reasoning
BioASQ [11]	En	–	Yes	Yes	Yes	No	need to refer web-pages (docs + concepts)
DocCVQA [18]	En	–	Yes	Yes	Yes	No	only 20 questions

Table 1: Document or web VQA datasets