Abstract

We present a new dataset for Visual Question Answering on document images called DocVQA. The dataset consists of 50,000 questions defined on 12,000+ document images. We provide detailed analysis of the dataset in comparison with similar datasets for VQA and reading comprehension. We report several baseline results by adopting existing VQA and reading comprehension models. Although the existing models perform reasonably well on certain types of questions, there is large performance gap compared to human performance (94.36% accuracy). The models need to improve specifically on questions where understanding structure of the document in crucial.

1. Introduction

Research in document analysis and recognition (DAR) is typically focused on generic information extraction tasks from document images that aim to convert imagery information into machine readable form, such as character recognition [10], table extraction [22] or key-value pair extraction [28]. Such algorithms tend to be designed as generic blocks, blind to the end-purpose the extracted information will be used for.

Progressing independently in such information extraction processes has been quite successful, although it is not necessarily true that holistic document image understanding can be achieved through a simple constructionist approach, building upon such modules. The scale and complexity of the task introduce difficulties that require a different point of view.

In this article we introduce Document Visual Question Answering (DocVQA), as a high-level task dynamically driving DAR algorithms to conditionally interpret document images. By doing so, we seek to inspire a purpose-driven point of view in DAR research. In case of Document VQA, as illustrated in Figure 1, an intelligent reading system is expected to respond to ad-hoc requests for information, as expressed in natural language questions by human users. To do so, reading systems should not only extract and interpret the textual (handwritten, typewritten or printed) content of the document images, but exploit numerous other visual cues including layout (page structure, forms, tables), non-textual elements (marks, tick boxes, separators, diagrams) and style (font, colours, highlighting), to mention just a few.

Departing from generic VQA [13] and Scene Text VQA [33, 5] approaches, the nature of document images requires a different approach to exploit all the above visual cues, making use of prior knowledge of the implicit written communication conventions used, and dealing with the high-density semantic information conveyed in such images. Answers cannot be sourced from a closed dictionary, but they are inherently open ended.

Previous approaches on bringing visual question answering to the documents domain have either focused on specific
document elements such as data visualisations [19, 21] or on specific collections such as book covers [26]. In contrast to such approaches, we recast the problem to its generic form, and put forward a large scale, varied collection of real documents.

The main contributions of this work can be summarized as following:

- We introduce DocVQA, a large scale dataset of 12,767 document images of varied types and content, over which we have defined 50,000 questions and answers. The questions defined are categorised based on their reasoning requirements, allowing us to perform detailed analysis of DocVQA methods.
- We define and evaluate various baseline methods over the DocVQA dataset, ranging from simple heuristic methods and human performance analysis that allow us to define upper performance bounds given different assumptions, to state of the art Scene Text VQA models and NLP models.

The DocVQA dataset and related code is publicly available to download and the challenge is open for continuous submission at the Robust Reading Competition (RRC) portal.

2. Related Datasets and Tasks

Machine reading comprehension (MRC) and open-domain question answering (QA) are two problems which are being actively pursued by Natural Language Processing (NLP) and Information Retrieval (IR) communities. In MRC the task is to answer a natural language question given a question and a paragraph or a single document as the context. In case of open domain QA, no specific context is given and answer need to be found from a large collection (say Wikipedia) or from Web. MRC is often modelled as an extractive QA problem where answer is defined as a span of the context on which the question is defined. Examples of datasets for extractive QA include SQuAD 1.1 [30], NewsQA [35] and Natural Questions [25]. MS MARCO [27] is an example of a QA dataset for abstractive QA, that is, answers need to be generated not extracted. Recently Transformer based pretraining methods like BERT [9] and XLNet [39] have helped to build QA models outperforming Humans on reading comprehension on SQuAD [30]. In contrast to QA in NLP where context is given as computer readable strings,

Visual Question Answering (VQA) aims to provide a correct answer given an image and a natural language question. VQA has attracted an intense research effort over the past few years [13, 1, 17]. Out of a large body of work on VQA, the scene text VQA branch is the most related to our work. Scene text VQA refers to VQA systems aiming to deal with cases where understanding scene instances is necessary to respond to the questions posed. The ST-VQA [5] and TextVQA [33] datasets were introduced in parallel in 2019 and were quickly followed by more research [34, 11, 37].

The ST-VQA dataset [5] has 31,000+ questions over 23,000+ images collected from different public data sets. The TextVQA dataset [33] has 45,000+ questions over 28,000+ images sampled from specific categories of the OpenImages dataset that are expected to contain text. Another dataset named OCR-VQA [26] comprises more than 1 million question-answer pairs over 207K+ images of book covers. The questions in this dataset are domain specific, generated based on template questions and answers extracted from available metadata. Certain effort has been made to paraphrase questions to gain some diversity. More than 50% of the questions have answers that are not scene text instances, including 40% binary (yes/no) questions and 10% questions about book genres for example.

Scene text VQA methods [16, 11, 33, 12] typically make use of pointer mechanisms in order to deal with out-of-vocabulary words appearing in the image and provide the open answer space required. This goes hand in hand with the use of word embeddings capable of encoding OOV words into a pre-defined semantic space, such as FastText [6] or BERT [9]. More recent, top-performing methods in this space include M4C [16] and MM-GNN [11] models.

Parallelly there have been works on certain domain specific QA tasks which require to read and understand text in the images. The DVQA dataset presented by Kafle et al. [20, 19] comprises synthetically generated images of bar charts and template questions defined automatically based on the bar chart metadata. The dataset contains more than three million question/answer pairs over 300,000 images.

FigureQA [21] comprises over one million yes or no style questions, grounded in over 100,000 images. Three different types of charts are used: bar, pie and line charts. Similar to DVQA, images and question-answer pairs are synthetically generated using template questions. Another related QA task is Textbook Question Answering (TQA) [23] which aims at answering multimodal questions given a context of text, diagrams and images. Here textual information is provided in computer readable format.

Compared to these existing datasets either concerning VQA on real word images, or domain specific VQA for charts or book covers, the proposed DocVQA comprise of document images. The dataset covers a multitude of different document types that include elements like tables, forms and figures, as well as a range of different textual, graphical and structural elements.
3. DocVQA

In this section we explain data collection and annotation process and present statistics and analysis of DocVQA.

3.1. Data Collection

Document Images: Images in the dataset are sourced from documents in UCSF Industry Documents Library\(^2\). The documents are organized under different industries and further under different collections. We downloaded documents from different collections and hand picked pages from these documents for use in the dataset. Majority of the documents in the library are binarized and the binarization has taken on a toll on the document quality. Hence we tried to minimize binarized pages since we did not want poor image quality to be a bottleneck for VQA.

We also prioritized pages with tables, forms, lists and figures over pages which only have running text.

The final set of images in the dataset are drawn from pages of 6,071 industry documents. We made use of documents from as early as 1900 to as recent as 2018. (Figure 2b). Most of the documents are from the 1960-2000 period and they include typewritten, printed, handwritten and born-digital text. There are documents from all 5 major industries for which the library hosts documents — tobacco, food, drug, fossil fuel and chemical. We use many documents from food and nutrition related collections, as they have a good number of non-binarized images. See Figure 2a for industry wise distribution of the 6071 documents used. The documents comprise a wide variety of document types as shown in Figure 2c.

Questions and Answers: Questions and answers on the selected document images are collected with the help of remote workers, using a Web based annotation tool.

The annotation process was organized in three stages. In stage 1, workers were shown a document image and asked to define at most 10 question-answer pairs on it. We encouraged the workers to add more than one ground truth answer per question in the cases where it is warranted.

Workers were instructed to ask questions which can be answered using text present in the image and to enter the answer verbatim from the document. This makes VQA on the DocVQA dataset an extractive QA problem similar to extractive QA tasks in NLP [30, 35] and VQA in case of ST-VQA [5]. The second annotation stage aims to verify the data collected in the first stage. Here a worker was shown an image and questions defined on it in the first stage (but not the answers from the first stage), and was required to enter answers for the questions. In this stage workers were also required to assign one or more question types to each question. The different question types DocVQA are discussed in subsection 3.2. During this second stage, if the worker finds a question to be inapt — language issues, ambiguity, no definite answer etc., an option to flag the question was provided. Such questions are not included in the dataset.

If none of the answers entered in the first stage match exactly with any of the answers from the second stage, the particular question is sent for review in a third stage. Here questions and answers are editable and the reviewer either accepts the question-answer (after editing if necessary) or ignores it. The third stage review is done by the authors themselves. Screen grabs of the three stages can be found.
3.2. Statistics and Analysis

The DocVQA comprises 50,000 questions framed on 12,767 images. The data is split randomly in an 80–10–10 ratio to train, validation and test splits. The train split has 39,463 questions and 10,194 images, the validation split has 5,349 questions and 1,286 images and the test split has 5,188 questions and 1,287 images.

As mentioned before, questions are tagged with question type(s) during the second stage of the annotation process. Figure 3 shows the 9 question types and percentage of questions under each type. A question type signifies the type of data where the question is grounded. For example, ‘tablelist’ is assigned if answering the question requires understanding of a table or a list. If the information is in the form of a key:value, the ‘form’ type is assigned. ‘Layout’ is assigned for questions which require spatial/layout information to find the answer. For example, questions asking for title, require one to understand structure of the document. If answer for a question is based on information in the form of sentences/paragraphs type assigned is ‘running text’. For all questions where answer is based on handwritten text, ‘handwritten’ type is assigned. Note that a question can have more than one type associated with it. (Examples from DocVQA for each question type are given in Appendix B.)

In the following analysis we compare statistics of questions, answers and OCR tokens with other similar datasets for vqa — VQA 2.0 [13], TextVQA [33] and ST-VQA [5] and SQuAD 1.1 [30] reading comprehension dataset. Statistics for other datasets are computed based on their publicly available data splits. For statistics on OCR tokens, for DocVQA we use OCR tokens generated by a commercial OCR solution. For VQA 2.0, TextVQA and ST-VQA we use OCR tokens made available by the authors of LoRRA [33] and M4C [16] as part of the MMF [32] framework.

Figure 4d shows the distribution of question lengths for questions in DocVQA compared with other similar datasets. The average question length is is 8.12, which is second highest among the compared datasets. In DocVQA 35,362 (70.72\%) questions are unique. Figure 4a shows the top 15 most frequent questions and their frequencies. There are questions repeatedly being asked about dates, titles and page numbers. A sunburst of first 4 words of questions is shown in Figure 6. It can be seen that a large majority of questions start with “what is the”, asking for date, title, total, amount or name.

Figure 5: Word cloud of words of answers (left) and word cloud of words recognized from the document images in the dataset (right)
Distribution of answer lengths is shown in Figure 4e. We observe in the figure that both DocVQA and SQuAD 1.1 have a higher number of longer answers compared to the VQA datasets. The average answer length is 2.17. 63.2% of the answers are unique, which is second only to SQuAD 1.1 (72.5%). The top 15 answers in the dataset are shown in Figure 4b. We observe that almost all of the top answers are numeric values, which is expected since there are a good number of document images of reports and invoices. In Figure 4c we show the top 15 non-numeric answers. These include named entities such as names of people, names of institutions and names of places. The word cloud on the left in Figure 5 shows frequent words in answers. Most common words are names of people and names of calendar months.

In Figure 4f we show the number of images (or ‘context’ in case of SQuAD 1.1) containing a particular number of text tokens. The average number of text tokens in an image or context is the highest in the case of DocVQA (182.75). It is considerably higher compared to SQuAD 1.1 where contexts are usually small paragraphs whose average length is 117.23. In case of VQA datasets which comprise real world images the average number of OCR tokens is not more than 13. The word cloud on the right in Figure 5 shows the most common words spotted by the OCR on the images in DocVQA. We observe that there is high overlap between common OCR tokens and words in answers.

4. Baselines

In this section we explain the baselines we use, including heuristics and trained models.

4.1. Heuristics and Upper Bounds

The heuristics we evaluate are: (i) Random answer: measures performance when we pick a random answer from the answers of the train split. (ii) Random OCR token: performance when a random OCR token from the given document image is picked as the answer. (iii) Longest OCR token is the case when the longest OCR token in the given document is selected as the answer. (iv) Majority answer measures the performance when the most frequent answer in the train split is considered as the answer.

We also compute the following upper bounds: (i) Vocab UB: This upper bound measures performance upper bound one can get by predicting correct answers for the questions, provided the correct answer is present in a vocabulary of answers, comprising all answers which occur more than once in the train split. (ii) OCR substring UB: is the upper bound on predicting the correct answer provided the answer can be found as a substring in the sequence of OCR tokens. The sequence is made by serializing the OCR tokens recognized in the documents as a sequence separated by space, in top-left to bottom-right order. (iii) OCR subsequence UB: upper bound of predicting the correct answer, provided the answer is a subsequence of the OCR tokens’ sequence.

4.2. VQA Models

For evaluating performance of VQA models on DocVQA we employ two models which have the capability to read text present in the images - Look, Read, Reason & Answer (LoRRA) [33] and Multimodal Multi-Copy Mesh (M4C) [16].

LoRRA: follows a bottom-up and top-down attention [3] scheme with additional bottom-up attention over OCR tokens from the images. In LoRRA, tokens in a question are first embedded using a pre-trained embedding (GloVe [29]) and then these tokens are iteratively encoded using an LSTM [15] encoder. The model uses two types of spatial features to represent the visual information from the images - (i) grid convolutional features from a Resnet-152 [14] which is pre-trained on ImageNet [8] and (ii) features extracted from bounding box proposals from an object detection model — a Faster R-CNN [31] pre-trained on Visual Genome data [24]. OCR tokens from the image are embedded using a pre-trained word embedding (FastText [7]). An attention mechanism is used to compute an attention weighted average of the image features as well the OCR tokens’ embeddings. These averaged features are combined and fed into an output module. The final classification layer of the model, predicts an answer either from a fixed vocabulary (made from answers in train set) or copy an answer.
from a dynamic vocabulary which essentially is the list of OCR tokens in an image. Here the copy mechanism can copy with only one of the OCR tokens from the image. Consequently it cannot output an answer which is a combination of two or more OCR tokens.

**M4C**: uses a multimodal transformer and iterative answer prediction as its backbone to yield state-of-the-art results on TextVQA [33], ST-VQA [5] and OCR-VQA [26] datasets. Here tokens in questions are embedded using a pre-trained Bidirectional Encoder Representations from Transformers (BERT) model [9]. Images are represented using (i) appearance features of the objects detected using an object detection model — Faster-RCNN [31] pre-trained on Visual Genome [24] and (ii) location information - bounding box coordinates of the detected objects. Each OCR token recognized from the image is represented using (i) a pretrained word embedding (FastText [7]), (ii) appearance feature of the OCR token’s bounding box from the same Faster R-CNN which is used for appearance features of objects (iii) PHOC [2] representation of the token and (iv) bounding box coordinates of the OCR token. Then these feature representations of the three entities (question tokens, objects and OCR tokens) are projected to a common, learned embedding space. Then a stack of Transformer [36] layers are applied over these features in the common embedding space. The multi-head self attention in transformers enable both inter-entity and intra-entity attention. Finally, answers are predicted through iterative decoding in an auto-regressive manner. Here the fixed vocabulary used is made up of the most common answer words in the train split. Note that in this case the fixed vocabulary comprises of answer words, not answers itself as in the case of LoRRA. At each step in the decoding, the decoded word is either an OCR token from the image or a word from the fixed vocabulary of common answer words.

In our experiments we use the original LoRRA and M4C models and few variants of these models. Since images in DocVQA are document images and have a higher number of OCR tokens compared to real world images in typical VQA datasets, we try out larger dynamic vocabularies (i.e. more OCR tokens are considered from the images) for both LoRRA and M4C. Similarly for both models we evaluate performance when no fixed vocabulary is used. Since the notion of visual objects in real word images is not directly applicable in case of document images, we also try out variants of LoRRA and M4C by omitting the features of objects.

### 4.3. Reading Comprehension Models

In addition to the VQA models which can read text, we try out extractive question answering / reading comprehension models from NLP space. In particular, we use BERT [9] question answering models. BERT is a method of pre-training language representations from unla-

| | val ANLS | Acc. | test ANLS | Acc. |
|---|---|---|---|---|
| Human | - | - | 0.981 | 94.36 |
| Random answer | 0.003 | 0.00 | 0.003 | 0.00 |
| Random OCR token | 0.013 | 0.52 | 0.014 | 0.58 |
| Longest OCR token | 0.002 | 0.05 | 0.003 | 0.07 |
| Majority answer | 0.017 | 0.90 | 0.017 | 0.89 |
| Vocab UB | - | 31.31 | - | 33.78 |
| OCR substring UB | - | 85.64 | - | 87.00 |
| OCR subsequence UB | - | 76.37 | - | 77.00 |

Table 1: Evaluation of different heuristics and upper bounds. Predicting random answers or majority answer do not even yield 1% accuracy. Answers are a substring of the serialized OCR output in more than 85% of the cases.

5. Experiments

In this section we explain evaluation metrics and our experimental settings and report results of experiments.

#### 5.1. Evaluation Metrics

Two evaluation metrics we use are Average Normalized Levenshtein Similarity (ANLS) and Accuracy (Acc.). ANLS was originally proposed for evaluation of VQA on ST-VQA [4]. The Accuracy metric measures percentage of questions for which the predicted answer matches exactly with any of the target answers for the question.

#### 5.2. Experimental setup

For measuring human performance, we collect answers for all questions in test split, with help of 4 volunteers from our institution.

For all our experiments including heuristics and trained baselines, OCR tokens we use are extracted using a commercial OCR application. For the heuristics and upper bounds where a vocabulary of answers is used, the vocabulary comprises the 4,341 answers which occur more than once in the train split.

For LoRRA and M4C models we use official implementations available as part of the MMF framework [32]. The training settings and hyper parameters are the same as the ones reported in the original works. The fixed vocabulary we use for LoRRA is same as the vocabulary we use for computing vocabulary based heuristics and upper bounds. For M4C the fixed vocabulary we use is a vocabulary of the 5,000 most frequent words from the answers in the train split.
Table 2: Performance of the VQA models which are capable of reading text — LoRRA [33] and M4C [16]. Detection of visual objects and their features (bottom-up attention), which is a common practice in VQA is ineffective in case of DocVQA.

| Method   | Objects’ feature | Fixed vocab. | Dynamic vocab. size | val ANLS | Acc. | test ANLS | Acc. |
|----------|------------------|--------------|---------------------|----------|------|-----------|------|
| LoRRA    | ✓                | ✓            | 50                  | 0.110    | 7.22 | 0.112     | 7.63 |
|          | ✓                | ✗            | 50                  | 0.041    | 2.64 | 0.037     | 2.58 |
|          | ✗                | ✓            | 50                  | 0.102    | 6.73 | 0.100     | 6.43 |
|          | ✓                | ✓            | 150                 | 0.101    | 7.09 | 0.102     | 7.22 |
|          | ✓                | ✓            | 500                 | 0.094    | 6.41 | 0.095     | 6.31 |
| M4C      | ✓                | ✓            | 50                  | 0.292    | 18.34| 0.306     | 18.75|
|          | ✓                | ✗            | 50                  | 0.216    | 12.44| 0.219     | 12.15|
|          | ✗                | ✓            | 50                  | 0.294    | 18.75| 0.310     | 18.92|
|          | ✗                | ✓            | 150                 | 0.352    | 22.66| 0.360     | 22.35|
|          | ✓                | ✓            | 300                 | 0.367    | 23.99| 0.375     | 23.90|
|          | ✓                | ✓            | 500                 |          |      | 0.385     | 24.73|

Table 2: Performance of the VQA models which are capable of reading text — LoRRA [33] and M4C [16]. Detection of visual objects and their features (bottom-up attention), which is a common practice in VQA is ineffective in case of DocVQA.

For the BERT QA models we use three pre-trained BERT models from the Transformers library [38]. The models we use are bert-base-uncased, bert-large-uncased-whole-word-masking and bert-large-uncased-whole-word-masking-finetuned-squad. We abbreviate the model names as bert-base, bert-large and bert-large-squad respectively. Among these, bert-large-squad is a pre-trained model which is also finetuned on SQuAD 1.1 for question answering. Unlike extractive question answering or reading comprehension datasets, in DocVQA ‘contexts’ on which questions are asked, are the document images, not paragraphs of text. Hence to finetune the BERT QA models on DocVQA we need to prepare the data in SQuAD style format where the answer to a question is a ‘span’ of the context paragraph, defined by start and end indices of the answer. To this end first we serialize the OCR tokens recognized on the document images to a single string, separated by space, in top-left to bottom-right order. To approximate the answer spans we follow an approach proposed in TriviaQA [18], which is to find the first match of the answer string in the serialized OCR string.

The bert-base model is finetuned on DocVQA on 2 Nvidia GeForce 1080 Ti GPUs, for 2 epochs, with a batch size of 32. We use Adam optimizer with a learning rate of $5e^{-05}$. The bert-large and bert-large-squad models are finetuned on 4 GPUs for 6 epochs with a batch size of 8, and a learning rate of $2e^{-05}$.

5.3. Results

Results of all heuristic approaches and upper bounds are reported in Table 1. We can see that none of the heuristics get even a $1\%$ accuracy on the validation or test splits.

OCR substring UB yields 85.64 on validation and 87.00 on test set. This upper bound has a downside that the substring match in all cases needs not be an actual answer match. For example if the answer is “2” which is the most common answer in the dataset, it will match with a “2” in “2020” or a “2” in “2pac”. This is the reason why we evaluate the OCR substring UB. An answer is a sub sequence of the serialized OCR output in at least 76% of the cases in both validation and test splits.

Results of our trained VQA baselines are shown in Table 2. First rows for both the methods report results of the original model proposed by the respective authors. In the case of LoRRA the original model yields the best results compared to the variants of the model. With no fixed vocabulary, the performance of the model drops sharply suggesting that the model primarily relies on the fixed vocabulary to output answers. Increasing the dynamic vocabulary results in a slight performance drop suggesting that incorporating more OCR tokens from the document images does little help. Unlike the case of LoRRA, increasing the size of the dynamic vocabulary from 50 to 500 improves the ANLS by around 50% in both validation and test splits. And the variant which does not use features of visual objects performs slightly better than the original model.

Results of the BERT question answering models are reported in Table 3. We observe that all BERT models per-
Q: What is the underlined heading just above the table?
GT: Indications for implantation
M4C best: indications for implantation
BERT best: total aneurism
Human: indications for implantation

Q: What is the Extension Number as per the voucher?
GT: (910) 741-0673
M4C best: 963.12
BERT best: (910) 741-0673
Human: (910) 741-0673

Q: How many boxed illustrations are there?
GT: 9
M4C best: 4
BERT best: 4
Human: 9

Figure 7: Qualitative results from our experiments. The leftmost example is a ‘layout’ type question answered correctly by the M4C model but errored by the BERT model. In the second example the BERT model correctly answers a question on a form while the M4C model fails. In case of the rightmost example, both models fail to understand a step by step illustration on DocVQA. Answers predicted by this model match one of the target answers exactly, in ∼ 55% of the questions.

In Figure 8 we show performance by question type. We compare the best models among VQA models and BERT question answering models against the human performance on the test split. We observe that the human performance is uniform while the models’ performance vary for different question types. In Figure 7 we show a few qualitative results (more results in Appendix C) from our experiments.

6. Conclusion

We introduce a new data set and an associated VQA task with the aim to inspire a "purpose-driven" approach in document image analysis and recognition research. Our baselines and the initial results motivate the simultaneous use of visual and textual cues for answering questions asked on document images. This could drive methods that use the low-level cues (text, layout, arrangements) and high-level goals (purpose, relationship, domain knowledge) in solving problems of practical importance.
References

[1] Aishwarya Agrawal, Aniruddha Kembhavi, Dhruv Batra, and Devi Parikh. C-VQA: A compositional split of the visual question answering (VQA) v1.0 dataset. arXiv preprint arXiv:1704.08243, 2017.

[2] J. Almazn, A. Gordo, A. Forns, and E. Valveny. Word spotting and recognition with embedded attributes. TPAMI, 2014.

[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering, 2017.

[4] Ali Furkan Biten, Ruben Tito, Andrs Mafla, Luís Gmez, Marcl Rusniol, Miness Mathew, C. V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. ICDAR 2019 compeion on scene text visual question answering. CoRR, abs/1907.00490, 2019.

[5] Ali Furkan Biten, Ruben Tito, Andres Mafla, Luís Gómez, Marcl Rusiínol, Ernest Valveny, C. V. Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In ICCV, 2019.

[6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5.

[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 2017.

[8] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In CVPR, 2009.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In ACL, 2019.

[10] David Doermann, Karl Tombre, et al. Handbook of document image processing and recognition. Springer, 2014.

[11] DiFei Gao, Ke Li, Ruiping Wang, Shiguang Shan, and Xilin Chen. Multi-modal graph neural network for joint reasoning on vision and scene text. In CVPR, 2020.

[12] Luís Gómez, Ali Furkan Biten, Rubén Tito, Andrés Mafla, and Dimosthenis Karatzas. Multimodal grid features and cell pointers for scene text visual question answering. arXiv preprint arXiv:2006.00923, 2020.

[13] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Comput., 1997.

[16] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In CVPR, 2020.

[17] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In CVPR, pages 2901–2910, 2017.

[18] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In ACL, 2017.

[19] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In CVPR, 2018.

[20] Kushal Kafle, Robik Shrestha, Scott Cohen, Brian Price, and Christopher Kanan. Answering questions about data visualizations using efficient bimodal fusion. In WACV, 2020.

[21] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Akos Kadar, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning. arXiv preprint arXiv:1710.07300, 2017.

[22] Isaac Kavasidis, Carmelo Pino, Simone Palazzo, Francesco Rundo, Daniela Giordano, P Messina, and Concetto Spampinato. A saliency-based convolutional neural network for tabular and chart detection in digitized documents. In ICIAP, 2019.

[23] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In CVPR, 2017.

[24] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vision, 2017.

[25] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vision, 2017.

[26] Tri Nguyen et al. Ms marco: A human generated machine reading comprehension dataset. In ICDAR, 2017.

[27] Tri Nguyen et al. Ms marco: A human generated machine reading comprehension dataset. In ICDAR, 2017.

[28] Rasmus Berg Palm, Ole Winther, and Florian Laws. Cloudscan-a configuration-free invoice analysis system using recurrent neural networks. In ICDAR, 2017.

[29] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In EMNLP, 2014.

[30] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,00+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250, 2016.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In NeurIPS, 2015.
Amanpreet Singh, Vedanuj Goswami, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Mmf: A multimodal framework for vision and language research. https://github.com/facebookresearch/mmf, 2020.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In CVPR, 2019.

Ajeet Kumar Singh, Anand Mishra, Shashank Shekhar, and Anirban Chakraborty. From strings to things: Knowledge-enabled vqa model that can read and reason. In ICCV, 2019.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. CoRR, abs/1611.09830, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NeurIPS, 2017.

Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. On the general value of evidence, and bilingual scene-text visual question answering. In CVPR, 2020.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface’s transformers: State-of-the-art natural language processing. ArXiv, abs/1910.03771, 2019.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In NeurIPS, 2019.
A. Screen grabs of Annotation Tool

As mentioned in Section 3.1 in the main paper, annotation process involves three stages. In Figure A.1, Figure A.2 and Figure A.3 we show screen grabs from stage 1, stage 2 and stage 3 of the annotation process respectively.

B. Examples of Question Types

We define 9 question types, based on the kind of reasoning required to answer a question. Question types are assigned at the second stage of the annotation. We discuss the question types in Section 3.2. in the main paper.

Examples for types form, yes/no and layout are shown in Figure B.1. Examples for a question based on a handwritten date in a form (types form and handwritten) are shown in Figure B.2. An example for a question based on information in the form of sentences or paragraphs (type running text) is shown in Figure B.3. Examples for types photograph and table are shown in Figure B.4. An example for a question based on a plot (type figure) is shown in Figure B.5. In all examples a crop of the original image is shown below the original image, for better viewing of the image region where the question is based on.

C. Additional Qualitative Examples

Here we show more qualitative results from our baseline experiments. These results supplement the Results section (Section 5.3 ) in the main paper.

Remember that BERT [9] question answering model is designed to answer questions asked on sentences or paragraphs of text (reading comprehension). In Figure C.1 we show two examples where the model answers questions outside the ambit of reading comprehension style question answering. In Figure C.2 we show examples where the M4C [16] model outperforms the BERT model to answer questions based on text seen on pictures or photographs. Such questions are similar to questions in TextVQA [33] or ST-VQA [5] datasets where M4C model yield state-of-the-art results. In Figure C.3 we show an example where both the models yield inconsistent results when posed with questions of similar nature, highlighting lack of reasoning behind answering. In Figure C.4 we show two examples where both the M4C and BERT model fail to answer questions which require understanding of a figure or a diagram. In Figure C.5 we show how OCR errors have resulted in wrong answers although the models manage to ground the questions correctly.
Figure A.1: **Annotation stage 1 - Question Answer Collection:** Questions and answers are collected for a given document image. Annotator can add up to 10 questions for a document. The document can be skipped if it is not possible to frame questions on it.

Figure A.2: **Annotation stage 2 - Data Verification:** For each question shown annotators have to (i) enter answer(s) (answer(s) from first stage are not shown) and (ii) Tag the question with one or more question types from the 9 question types shown in a drop-down (question types assigned to a question are shown in green highlight color) or (iii) flag/ignore the question by selecting the check-box corresponding to one of the reasons such as “invalid question”, “Serious lang. issue” etc. (the reasons chosen for flagging a question are shown in red highlight color)
Figure A.3: **Annotation Stage 3: Reviewing answer mismatch cases**: If none of the answers entered in the first stage for a question match with any of the answers entered in the second stage, the question is sent for review in a third stage. This review is handled by the authors and reviewer is allowed to edit question as well answers or add new answers before accepting the question.
Q: Is it an existing item?
   Question types: form and yes/no
   A: yes

Q: What is the date given at the top left?
   Question types: layout
   A: 03/17/98

Figure B.1: On the left is a question based on an yes/no check box. On the right, the question seeks for a date given at a particular spatial location — top left of the page.
### Q: What is the date written next to RSM approval?

**Question types:** form and handwritten

**A:** 3-17-98

Figure B.2: Date is handwritten and it is shown in a key:value format.
Q: If the request needs to be warehoused by RJR, what needs to be done?
Question types: running text
A: write to RJR

Figure B.3: Question is grounded on a sentence.
Q: Whose picture is given?  
**Question types:** photograph and layout  
A: Dr. Dwayne G. Westfall

Q: What is the average sucrose % for N level 501+?  
**Question types:** table  
A: 15.9

Figure B.4: On the left is a question asking for name of the person in the photograph. To answer the question on the right, one needs to parse the table and pick the value in the appropriate cell.
Q: What is the highest value for “Intake, mg/1000kcal” plotted on the ‘X’ axis of the graph?

**Question types:** figure

**A:** 300

Figure B.5: Question is based on the plot shown at the bottom of the given image, asking for the highest value on the X axis
Q: What is the total cost for Fat cell size (Mt. Sinai) in the -05 year?
GT: $35,864
M4C best: 4400
BERT best: $35,864
Human: $35,864

Q: What is the first recipe on the page?
GT: hawaiian fruit cake
M4C best: island desserts (continued from cake)
BERT best: hawaiian fruit cake
Human: hawaiian fruit cake

Figure C.1: Examples where BERT QA model [9] answers questions other than ‘running text’ type. On the left is a question based on a table and for the other question one needs to know the ‘first recipe’ out of the two recipes shown. For the first question the model gets the answer correct except for an extra space, and in case of the second one the predicted answer matches exactly with the ground truth answer.
Figure C.2: How does the M4C [16] model perform on questions based on pictures or photographs. Here we show two examples where the best variant of the M4C model outperform the BERT best model in answering ‘layout’ type questions seeking to read what is written in a logo/pack. The BERT model doesn’t make any predictions for the questions.
Q: What was the committee strength for the first meeting?

GT: 6
M4C best: 6
BERT best: 6
Human: 6

Q: What was the committee strength for the last meeting?

GT: 5
M4C best: 6
BERT best: 6
Human: 5

Figure C.3: **Contrasting results for similar questions.** Here both the questions are based on the table at the bottom of the image. Both questions ask for ‘committee strength’ for a particular meeting (first or last). Both models get the answer right for the first one. But for the question on the right, the models predict same answer as the first one (“6”) while the ground truth is “5”. This suggests that the models’ predictions are not backed by a proper reasoning/grounding in all cases.
Q: What is the position above “vice chairman”?

GT: chairman
M4C best: legal counsel
BERT best: legal counsel
Human: chairman

Q: What is the highest value shown on the vertical axis?

GT: 99.99
M4C best: 50
BERT best: 32
Human: 99.99

Figure C.4: Understanding figures and diagrams. In case of the question on the left, one needs to understand an organizational hierarchy diagram. For the second question, one needs to know what a ‘vertical axis’ is, and then find the largest value. Both the models fail to answer the questions.
Figure C.5: **Impact of OCR errors.** Here the models are able to ground the questions correctly on the relevant information in the image, but failed to get the answers correct owing to the OCR errors. In case of the question on the left, even the answer entered by the human volunteer is not exactly matching with the ground truth. In case of the second question, OCR has split the date into multiple tokens due to over segmentation, resulting in incorrect answers by both the models.