InfographicVQA

Minesh Mathew[1] Viraj Bagal[2*] Rubèn Tito[3]
Dimosthenis Karatzas[3] Ernest Valveny[3] C.V. Jawahar[1]
[1]CVIT, IIIT Hyderabad, India [1]IISER Pune, India [3]Computer Vision Center, UAB, Spain
minesh.mathew@research.iiit.ac.in, viraj.bagal@students.iiserpune.ac.in, rperez@cvc.uab.cat

Abstract

Infographics communicate information using a combination of textual, graphical and visual elements. This work explores the automatic understanding of infographic images by using a Visual Question Answering technique. To this end, we present InfographicVQA, a new dataset comprising a diverse collection of infographics and question-answer annotations. The questions require methods that jointly reason over the document layout, textual content, graphical elements, and data visualizations. We curate the dataset with an emphasis on questions that require elementary reasoning and basic arithmetic skills. For VQA on the dataset, we evaluate two Transformer-based strong baselines. Both the baselines yield unsatisfactory results compared to near perfect human performance on the dataset. The results suggest that VQA on infographics—images that are designed to communicate information quickly and clearly to human brain—is ideal for benchmarking machine understanding of complex document images. The dataset is available for download at docvqa.org

1. Introduction

Infographics are documents created to convey information in a compact manner using a combination of textual and visual cues. The presence of the text, numbers and symbols, along with the semantics that arise from their relative placements, make infographics understanding a challenging problem. True document image understanding in this domain requires methods to jointly reason over the document layout, textual content, graphical elements, data visualisations, color schemes and visual art, among others. Motivated by the multimodal nature of infographics, and the human centered design, we propose a Visual Question Answering (VQA) approach to infographics understanding.

VQA received significant attention over the past few years [15, 5, 16, 20, 23, 3]. Several new VQA branches focus on images with text, such as answering questions by looking at text books [24], business documents [31], charts [21, 22, 10] and screenshots of web pages [41]. Still, infographics are unique in their combined use and purposeful arrangement of visual and textual elements.

How many companies have more than 10K delivery workers?
Answer: 2 Evidence: Figure
Answer-source: Non-extractive Operation: Counting Sorting

Who has better coverage in Toronto - Canada post or Amazon?
Answer: canada post Evidence: Text
Answer-source: Question-span Image-span Operation: *none*

In which cities did Canada Post get maximum media coverage?
Answer: vancouver, montreal Evidence: Text Map
Answer-source: Multi-span Operation: *none*

Figure 1: Example image from InfographicVQA along with questions and answers. For each question, source of the answer, type of evidence the answer is grounded on, and the discrete operations required to find the answer are shown.

In this work, we introduce a new dataset for VQA on infographics, InfographicVQA, comprising $30,035$ questions over $5,485$ images. An example from our dataset is shown in Figure 1. Questions in the dataset include questions grounded on tables, figures and visualizations and questions that require combining multiple cues. Since most infographics contain numerical data, we collect questions that require elementary reasoning skills such as counting,

*Work done during an internship at IIIT Hyderabad.

Dataset	Images	Synthetic Images	Template questions	Text type	# Images	# Questions	Answer type
TQA [24]	Science diagrams	✗	✗	MR	1K	26K	MCQ
RecipeQA [48]	Culinary pictures	✗	✓	MR	251K	37K	MCQ
ST-VQA [7]	Natural images	✗	✗	ST	23K	31K	Ex
TextVQA [39]	Natural images	✗	✗	ST	28K	45K	Ex, SAb
OCR-VQA [32]	Book covers	✗	✓	BD	207K	1M	Ex, Y/N
DVQA [21]	Bar charts	✓	✓	BD	300K	3.4M	Ex, Nm, Y/N
FigureQA [22]	Charts - 5 types	✓	✓	BD	120K	1.5M	Y/N
LEAF-QA [10]	Charts - 4 types	✓	✓	BD	250K	2M	Ex, Nm, Y/N
VisualMRC [41]	Webpage screenshots	✗	✗	BD	10K	30K	Ab
DocVQA [31]	Industry documents	✗	✗	Pr, Tw, Hw, BD	12K	50K	Ex
InfographicVQA	Infographics	✗	✗	BD	5.4K	30K	Ex, Nm

Table 1: **Summary of VQA and Multimodal QA datasets where text on the images needs to be read to answer questions.** Text type abbreviations are: Machine Readable: MR, Scene Text: ST,Born Digital: BD, Printed: Pr, Handwritten: Hw, and Typewritten: Tw. Answer type abbreviations are: Multiple Choice Question: MCQ, Extractive: Ex, Short abstractive: SAb, Abstractive: Ab, Yes/No: Y/N, and Numerical (answer is numerical and not extracted from image or question; but derived): Nm.

sorting and arithmetic operations. We believe our dataset is ideal for benchmarking progress of algorithms at the meeting point of vision, language and document understanding.

We adapt a multimodal Transformer [42]-based VQA model called M4C [19] and a layout-aware, BERT [12]-style extractive QA model called LayoutLM [46] for VQA on InfographicVQA. Results using these two strong baselines show that current state-of-the-art (SoTA) models for similar tasks perform poorly on the new dataset. The results also highlight the need to devise better feature extractors for infographics, different from bottom-up features [4] of visual 'objects' that are typically used for VQA on natural scene images.

2. Related works

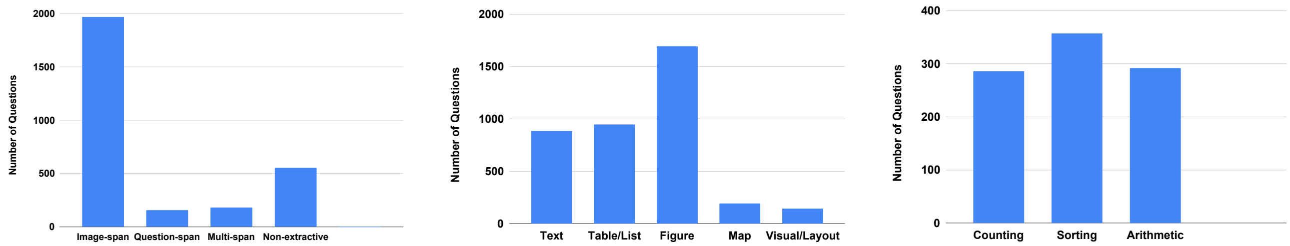
Question answering in a multimodal context. Textbook Question Answering (TQA) [24] and RecipeQA [48] deal with Question Answering (QA) in a multimodal context. For TQA, contexts are textbook lessons and for RecipeQA, contexts are recipes containing text and images. Contrary to InfographicVQA and other datasets mentioned below, text in these two datasets are not embedded on the images, but provided in machine-readable form, as a separate input.

ST-VQA [7] and TextVQA [39] datasets extend VQA over natural images to a new direction where understanding scene text on the images is necessary to answer the questions. While these datasets comprise images captured in the wild with sparse text content, InfographicVQA has born-digital images with an order of magnitude more text tokens per image, richer in layout and in the interplay between textual and visual elements. OCR-VQA [32] introduces a task similar to ST-VQA and TextVQA, but solely on images of book covers. Template questions are generated from book metadata such as author name and title. Consequently, question-answers in the dataset are less reliant on visual information. DVQA [21], FigureQA [22], and LEAF-QA [10] datasets deal with VQA on charts. All three datasets have chart images rendered using chart plotting libraries and template questions.

DocVQA [31] comprises images of pages from industry/business documents. Questions in the dataset are grounded on document elements such as passages, tables, forms and charts. Similar to ST-VQA, DocVQA is an extractive VQA task where answers can always be extracted verbatim from the text on the images. VisualMRC [41] on the other hand, is an abstractive VQA (answers cannot be directly extracted from text in the images or questions) benchmark where images are screenshots of web pages. Compared to VisualMRC, InfographicVQA is an extractive VQA task (answers are extracted as 'span'(s) of the question or text present in the given image), except for questions that require certain discrete operations resulting in numerical non-extractive answers. (see subsection 3.2). Table 1 presents a high-level summary of the QA/VQA datasets related to ours.

Multimodal transformer for Vision-Language tasks. Following the success of BERT [12]-like models for Natural Language Processing (NLP) tasks, there have been multiple works extending it to the Vision-Language space. Models like VL-BERT [40], VisualBERT [27], and UNITER [11] show that combined pretraining of BERT-like architectures on vision and language inputs achieve SoTA performances on various downstream tasks, including VQA on natural images. For VQA on images with scene text, M4C and TAP [49] use a multimodal transformer block to fuse embeddings of question, scene text tokens, and objects detected from an image.

The success of transformer-based models for text understanding inspired the use of similar models for document image understanding. LayoutLM and LAMBERT [14]



(a) Answer-sources and their counts

(b) Evidence types and their counts

(c) Operation types and their counts

Figure 2: Count of questions in validation set by their Answer-source, (2a), Evidence required to answer (2b) and the discrete Operation performed to find the answer (2c).

incorporate layout information to the BERT architecture by using embeddings of the 2D positions of the text tokens in the image. One of the strong baselines we use in this work is based on the LayoutLM model. Concurrent to this work, there have been multiple works published on arXiv that deal with a joint understanding of the text, image and layout in document images. Models such as LayoutLMv2 [47], TILT [34], DocFormer [6] and StrucText [28] build on transformer-based architectures and leverage large-scale pretraining on unlabelled data, using pretraining objectives specifically designed for document understanding.

Infographics understanding. Bylinskii *et al*. [9] and Madan *et al*. [30] looked at generating textual and visual tags from infographics. Landman uses an existing text summarization model to generate captions for infographics [26]. But the model uses only text recognized from infographics to generate the captions and layout/visual information is not considered. These three works use Visually29K dataset that comprises images from a single infographics website. MASSVIS [8] is a collection of infographics created to study infographics from a cognitive perspective. As observed by Lu *et al*. [29], it is a specialized collection focusing on illustrations of scientific procedures and statistical charts, therefore not representative of general infographics.

To summarize, existing datasets containing infographics are either specialized collections or infographics collected from a single source. In contrast, the InfographicVQA dataset comprises infographics drawn from thousands of different sources, with diverse layouts and designs, and without any topic specialization.

3. InfographicVQA

A brief description of the data collection and detailed analysis of the data is presented here. Refer to Section A in the supplementary material for more details on data collection.

3.1. Collecting images and question-answer pairs

Infographics in the dataset were downloaded from the Internet for the search query "infographics" . The downloaded images are cleaned for removal of duplicates before adding them to the annotation tool. Unlike crowd-sourced annotation, InfographicVQA was annotated by a small number of annotators using an internal annotation tool. The annotation process involved two stages. In the first stage, workers were required to add question-answer pairs based on an infographic shown to them. Similar to the SQuAD dataset [35], to make the evaluation more robust, an additional answer was collected for each question in the validation and test split during the second stage of annotation. At this stage, workers were shown an image annotated in the first stage along with the questions asked on it. They were instructed to answer the questions or flag a question if it was unanswerable.

3.2. Question-answer types: answer-source, evidence and operation

In the second stage, in addition to answering questions collected in the first stage, we instructed the workers to add question-answer types (QA types). QA types are a set of category labels assigned to each question-answer pair. DocVQA and VisualMRC have QA types that indicate the kind of document object a question is based on. DROP [13] dataset for reading comprehension defines answer types such as question span and passage span and categorizes questions by the kind of discrete operations arithmetic or logical operations required to find the answer. In InfographicVQA we collect QA types under three categories — Answer-source, Evidence and Operation.

There are four types of Answer-source — Image-span, Question-span, Multi-span and Non-extractive. Akin to the definition of 'span' in SQuAD [35] and DocVQA, an answer is considered Image-span if it corresponds to a single span (a sequence of text tokens) of text, extracted verbatim, in the reading order, from text present in the image. Similarly, when the answer is a span from the question it is labelled as Question-span. In Figure 1, answer to the second question is found both in the image and question as single sequence of contiguous tokens (or a 'span'). Hence there are two answer sources for the question — Image-span and Question-span. A Multi-span answer is composed of multiple spans of text from the image. Like the DROP dataset annotation, we instructed our workers to enter Multi-span answers by separating each individual span by a comma and a white space. For example, in Figure 1 for the last question the answer is names of two cities, which do not appear in a contiguous sequence of text. Hence it is a Multi-span answer. For a Multi-span answer, any order of the individual spans is a valid answer. In the above example, both "Vancouver, Montreal" and "Montreal, Vancouver" are valid answers. Since such answers are unordered lists, we consider all permutations of the list as valid answers for the question at evaluation time. The 'Non-extractive' type is assigned when the answer is not an extracted one. While collecting question-answer pairs, Non-extractive questions were allowed only if the answer is a numerical value. Inclusion of Question-span, Multi-span and numerical Non-extractive answers in InfographicVQA is inspired by a similar setting in the DROP dataset. We see this as a natural next step in VQA involving text, different from the purely extractive QA setting in datasets like DocVQA and ST-VQA, and abstractive question answering in VisualMRC where automated evaluation is difficult. Allowing only numerical answers in the non-extractive case makes sure that such answers are short and unique, giving no room for variability. Near perfect human performance while using automatic evaluation metrics (Table 4) validates that answers in InfographicVQA are unique with minimal variability when answered by different individuals.

The Evidence type indicates the kind of evidence behind the answer. Types of evidence are Text, Figure, Table/List, Map and Visual/Layout. For example, Map is used if the question is based on data shown on a geographical map. Visual/Layout type is added when evidence is based on the visual or layout aspect of the image. For example, questions such as "What is the color of the hat - brown or black?" or "What is written at the top left corner" fall in this category. Sometimes it is difficult to discern evidence for a question-answer pair. For example, for the first question in Figure 1, although the evidence type assigned by the worker is 'Figure', it could even be 'Table/List' since the visualization looks like a table. The operation type captures the kind of discrete operation(s) required to arrive at an answer — Counting, Arithmetic or Sorting.

Figure 2 shows the distribution of questions in the validation split based on Answer-source, Evidence and Operation. As evident from Figure 1, a question can have multiple types of answer source, evidence, or operation and many questions do not require any of the specified discrete operations to find the answer. For these reasons, counts in plots shown in Figure 2 do not add up to 100%.

3.3. Summary, statistics, and analysis

Dataset	Questions %Unique	Questions Avg. len	Answers %Unique	Answers Avg. len	Avg. tokens per image
ST-VQA	84.84	8.80	65.63	1.56	7.52
TextVQA	80.36	8.12	51.74	1.51	12.17
VisualMRC	96.26	10.55	91.82	9.55	151.46
DocVQA	72.34	9.49	64.29	2.43	182.75
InfographicVQA	99.11	11.54	48.84	1.60	217.89

Table 2: Statistics of questions, answers and OCR tokens in InfographicVQA and other similar VQA datasets.

InfographicVQA dataset has $30,035$ questions and $5,485$ images in total. These images are from $2,594$ distinct web domains. The data is split randomly to $23,946$ questions and $4,406$ images in train, $2,801$ questions and 500 images in validation, and $3,288$ questions and 579 images in test splits. We show basic statistics of questions, answers and OCR tokens in InfographicVQA and other similar datasets in Table 2.

Questions. Table 2 shows that InfographicVQA has the highest percentage of unique questions and the highest average question length compared to similar datasets. Figure 3 shows a sunburst of the common questions in the dataset. There are a good number of questions asking for "How many..." or percentages. This is expected since infographics carry a lot of numerical data.

Answers. It can be seen in Figure 4 that the most common answers are numbers. This is the reason why InfographicVQA has a smaller number of unique answers and smaller average answer lengths.

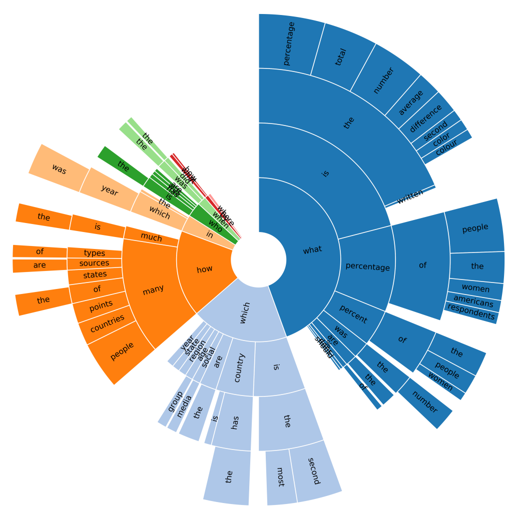


Figure 3: Staring 4-grams of common questions in InfographicVQA.

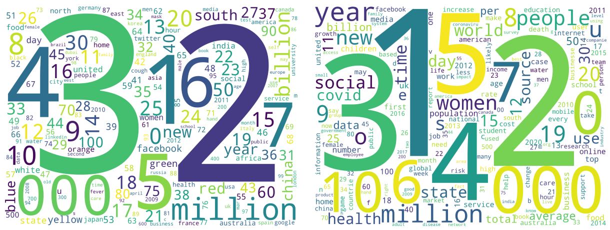


Figure 4: Word cloud of answers (left) and word cloud of words recognized from the infographics (right).

Embedded text. The number of average text tokens in InfographicVQA images is 217.89. Word cloud of Optical Character Recognition (OCR) tokens spotted on the images is shown in the wordcloud on the right in Figure 4. Common answers and common text tokens in images are similar.

4. Baselines

In this section, we describe the baselines we evaluated on the InfographicVQA. These include heuristic baselines and upper bounds, and SoTA models for VQA and document image understanding.

4.1. Heuristic baselines and upper bounds

Heuristic baselines and upper bounds we evaluate are similar to the ones evaluated in other VQA benchmarks like TextVQA and DocVQA.

Heuristic Baselines. We evaluate performance when an answer predicted for each questions is i) **Random answer** from the train split, ii) **Majority answer** from the train split and iii) **Random OCR token** from the image on which question is asked.

Upper bounds. We evaluate the performance upper bound on predicting the correct answer if the answer is present in a vocabulary of the most common answers in train split. This upper bound is called **Vocab UB**. Following DocVQA, to assess the percentage of questions for which answers can be found from the given OCR transcriptions, we compute **OCR UB** that measures the upper bound on performance if we always predict the correct answer, provided the answer is a sub sequence of the serialized OCR transcription of the given infographic. We serialize OCR tokens in the natural reading order, i.e., from top-left to bottom-right. **Vocab+OCR UB** is the percentage of questions that are either in Vocab UB or OCR UB.

4.2. M4C

M4C uses a Transformer stack to fuse representations of a question, OCR tokens, and image. Answers are predicted using an iterative, auto-regressive decoder decoding one word at a time, either from a fixed vocabulary or from the OCR tokens spotted on the image. Original M4C uses Region of Interest (ROI) pooled features from Box head of a Faster-RCNN [36] as the bottom-up visual features. Visual features of both the objects detected on the image and the OCR tokens spotted on the image are used.

4.3. LayoutLM

LayoutLM [46] extends BERT by incorporating layout information into the original BERT model. LayoutLM pretrained on millions of document images has proven to be effective for multiple document image understanding tasks. To adapt LayoutLM for InfographicVQA, we change the input to suit a multimodal setting and use an output head 'span' prediction. Since we are using SQuAD-style span prediction at the output, this model can only handle questions whose Answer-source is Image-span. Nearly 70% of questions in validation and test split are of this type. Extending this model to include questions with other Answer-sources is an interesting direction for future work.

4.3.1 Model overview

A schematic of the LayoutLM-based model which we use for InfographicVQA is shown in Figure 5. The input sequence to the model starts with a special [CLS] token, followed by the question tokens and the OCR tokens. The sequence ends with a special [SEP] token. Question and OCR tokens are also separated by a [SEP] token. All the tokens in the input sequence are represented by a corresponding embedding which in turn is the sum of i) a token embedding, ii) a segment embedding, iii) a 1D position embedding, iv) Four 2D position embeddings, and v) a visual embedding.

Following the original setting in BERT, for **token embedding,** we use WordPiece embeddings [45] with a $30,000$ size vocabulary. These embeddings are of size 768 . A **segment embedding** differentiates different segments in the input. For example, when the input sequence is formed by tokens from question and OCR tokens, question tokens and OCR tokens are given a segment id of 0 and 1, respectively. **1D position embedding** is used to indicate the order of a token within the sequence. For OCR tokens we use the default reading order by serializing the tokens in top-left to bottom-right order.

For 2D position embedding, we follow the same process as original LayoutLM. Given the bounding box coordinates (x_1, y_1, x_2, y_2) of an OCR token, we embed all the 4 coordinate values using 4 separate embedding layers. x_1 and x_2 share same embedding table and y_1 and y_2 share another common embedding table. The coordinate values are normalized to lie in the range 0–1000 before embedding. For [CLS] token we use 2D embedding corresponding to $(0, 0, 1000, 1000)$. For question tokens all the four 2D position embeddings used are 0s.

Unlike in original LayoutLM where visual features are fused after getting the attended embeddings from the Transformer block, we fuse the visual features early with the text

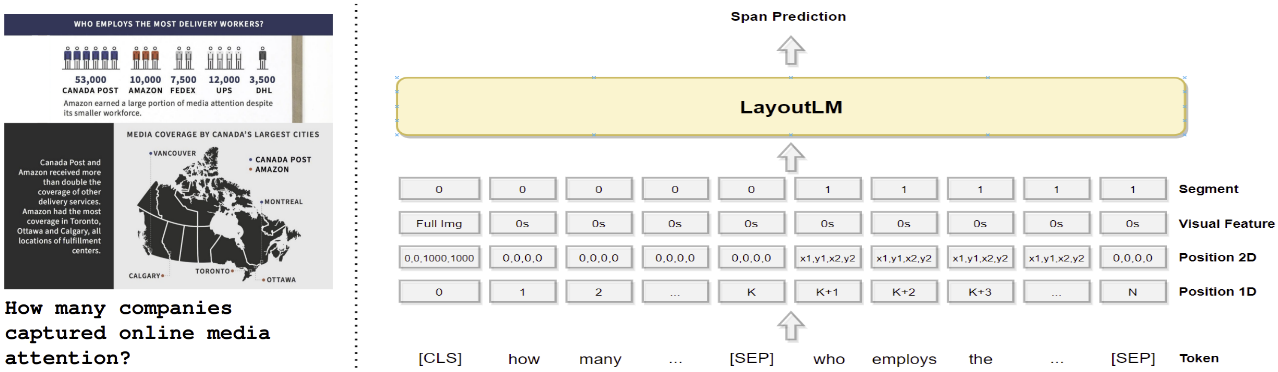


Figure 5: Overview of our LayoutLM based model for predicting answer spans. Textual, visual and layout modalities are embedded and mapped to the same space. Then, they are added and passed as input to a stack of Transformer layers.

input. Similar to M4C, for each OCR token, we use ROI pooled feature from the Box head of a pretrained object detection style model. This feature is mapped to the same size as other embeddings using a linear projection layer. For [CLS] and other special tokens we add visual feature corresponding to an ROI covering the entire image, named as "Full Img" in Figure 5.

4.3.2 Training procedure

Similar to the BERT and original LayoutLM, we train the model in two stages.

Pretraining: Following original LayoutLM, we use Masked Visual-Language Model (MVLM) task for pretraining with a masking probability of 0.15. Whenever masking, we replace each token with the [MASK] token 80% of the time, with a random token 10% of the time and keep it unchanged 10% of the time.

Finetuning: For finetuning, similar to BERT QA model for SQuAD benchmark, we use an output head that predicts start and end token positions of the answer span.

5. Experiments and results

In this section we report the experimental setting and results.

Detector	TextVQA		DocVQA		InfographicVQA	
	Avg.	<2 det.(%)	Avg.	<2 det.(%)	Avg.	<2 det.(%)
VG	28.8	0.0	4.1	43.9	7.4	23.9
DLA	1.0	97.9	4.7	0.0	2.9	43.4

Table 3: Statistics of object detections using two detectors – VG and DLA. DLA is trained for detecting document layout objects and VG is an object detection model trained on Visual Genome. Avg. shows average number of detections per image. ' <2 det.(%)' is the percentage of images on which number of detected objects is less than 2.

Baseline	ANLS		Accuracy(%)	
	val	test.	val	test
Human performance	-	0.980	-	95.70
Random answer	0.006	0.005	0.00	0.00
Random OCR token	0.011	0.014	0.29	0.49
Majority answer	0.041	0.035	2.21	1.73
Vocab UB	-	-	53.16	51.34
OCR UB	-	-	53.95	56.96
Vocab + OCR UB	-	-	76.71	77.4

Table 4: Results of heuristics and upper bounds. Heuristics yield near zero results. More than 75% of the questions have their answer present either in a fixed vocabulary or as an Image-span of the OCR tokens serialized in default reading order.

5.1. Experimental setup

Evaluation metrics. For evaluating VQA performance on InfographicVQA, we use Average Normalized Levenshtein Similarity (ANLS) and Accuracy metrics. The evaluation setup is same as the evaluation in DocVQA.

OCR transcription. Text transcriptions and bounding boxes for text tokens in the images are obtained using Textract OCR [1].

Human performance For evaluating human performance, all questions in the test split of the dataset are answered with the help of two volunteers (each question answered by a single volunteer).

Vocabulary of most common answers. For Vocab UB and heuristics involving a vocabulary, we use a vocabulary of 5,000 most common answers in the train split.

ROI Features. For our experiments using M4C and LayoutLM models, visual features of different bounding regions from the images are used. To this end, we use two pretrained object detection models — a Faster-RCNN [36] trained on Visual Genome [25] and a Mask-RCNN [17] trained on document images in PubLayNet [50] for Document Layout Analysis (DLA). We refer to these detectors as VG and DLA, respectively, in further discussions. The

FasterRCNN model we use is same as the one used for M4C. We use the implementation in MMF framework [37]. The DLA detector we use is from a publicly available Detectron2 [44]-based implementation [18]. Features from the last or second last Fully Connected (FC) layer are used as visual features in M4C and LayoutLM model. In VG and DLA, these features are of size 2048 and 1024 respectively.

In Table 3 we summarize the results while using the two detectors on TextVQA, DocVQA and InfographicVQA. With DLA, we notice that many of its detections, especially when there is only one detection per image is a box covering the entire image.

Experimental setting for M4C. We use the official implementation of the model [37]. The training parameters and other implementation details are the same as the ones used in the original paper. As done in original M4C, fixed vocabulary used with the model is created from $5,000$ most common words among words from answers in the train split.

Experimental setting for LayoutLM. The model is implemented in Pytorch [33]. In all our experiments, we start from a pretrained checkpoint of LayoutLM model made available by the authors in Hugginface's Transformers model zoo [43, 2]. The newly introduced linear projection layer which maps the ROI pooled features to the common embedding size of 768, is initialized from scratch. The features are from the last FC layer of the Box head of DLA or VG. To continue pretraining using in-domain data, we use four samples in one batch and Adam optimizer with a learning rate $2e-5$. For finetuning, we use a batch size of 8 and Adam optimizer with learning rate $1e-5$. For in-domain pretraining and finetuning no additional data other than train split of InfographicVQA is used. To map answers in InfographicVQA train split to SQUAD [35]-style spans, we follow the same approach used by Mathew *et al*. for DocVQA. We take the first subsequence match of an answer in the serialized transcription as the corresponding answer span. This way we find approximate spans for 52% of questions in the train split. Rest of the questions are not used for finetuning the model.

5.2. Results

Results of heuristic baselines, upper bounds, and human performance are shown in Table 4. Human performance is comparable to the human performance on DocVQA. As given by the Vocab + OCR UB, more than three quarters of questions have their answers present as a span of the OCR tokens serialized in the default reading order or in a vocabulary of most common answers in the train split.

We show results using M4C model in Table 5. In contrast to the original setting for which finetuning of visual features and features of detected objects are used, a setting that uses no finetuning and only a single visual feature corresponding to ROI covering the entire image, yields the best result.

Visual Feature	Finetune detector	Object& Count	# OCR tokens	ANLS val	ANLS test	Accuracy(%) val	Accuracy(%) test
VG	✓	Obj. (100)	50	0.107	0.119	4.81	4.87
VG	✓	Obj. (20)	50	0.111	0.122	4.82	4.87
VG	✗	Obj. (20)	50	0.125	0.127	4.89	4.89
VG	✗	Obj. (20)	300	0.128	0.134	4.90	5.08
VG	✗	None	300	0.136	0.143	5.86	6.58
VG	✗	Full Img	300	**0.142**	**0.147**	5.93	6.64
DLA	✗	Obj. (20)	50	0.110	0.130	4.86	5.02
DLA	✗	Obj. (20)	300	0.132	0.144	5.95	6.50
DLA	✗	None	300	0.140	0.142	5.90	6.39
DLA	✗	Full Img	300	0.138	0.140	5.97	6.42

Table 5: Performance of different variants of the M4C model. The original M4C setting is the one shown in the first row. 'Finetune detector' denotes the case when features from penultimate FC layer is used and last FC layer is finetuned along with the M4C model. This is the default setting in M4C. In our experiments, we get better results without finetuning. 'Obj. (100)' is the case when features from up to 100 objects (bottom-up features) are used. We experiment with 20 objects per image and the results did not change much. Using no object ('None') and feature from only one object—a box covering the entire image ('Full Img')—yield better results than the case where bottom-up objects are used.

Results of the LayoutLM based model are shown in Table 6. In-domain pretraining, using text from question, and OCR tokens help the model significantly. This is inline with observation by Singh *et al*. that pretraining on data similar to the data for a downstream task is highly beneficial in visio-linguistic pretraining [38]. The model that uses Full Img feature from DLA, added to the CLS performs the best on validation set. On the test set, a model which does not use any visual feature performs the best.

From Table 6, it is evident that models which use visual features of OCR tokens do not give better results. This im-

Full Img to	Visual feature	Continue pretrain.	OCR visual	ANLS val	ANLS test	Accuracy (%) val	Accuracy (%) test
-	-	✗	✗	0.212	0.225	13.40	15.32
-	-	✓	✗	0.250	**0.272**	18.14	19.74
CLS	DLA	✓	✗	**0.256**	0.261	18.56	19.16
All	DLA	✓	✗	0.248	0.266	17.82	18.77
Non-OCR	DLA	✓	✓	0.245	0.263	17.21	18.37
CLS	VG	✓	✗	0.229	0.235	16.47	16.51
All	VG	✓	✗	0.109	0.106	5.43	4.96
Non-OCR	VG	✓	✓	0.042	0.037	1.75	1.28

Table 6: Performance of LayoutLM with different input settings. Row 1 and 2 show LayoutLM's performance with and without in-domain pretraining. 'Visual Feature' column specify the kind of detector used for visual feature. 'OCR visual' indicate whether visual features of the OCR tokens are used or not. 'CLS', 'All' and 'Non-OCR' in 'Full Img to' column represent Full Img feature added only to CLS token, all tokens and all non OCR tokens respectively.

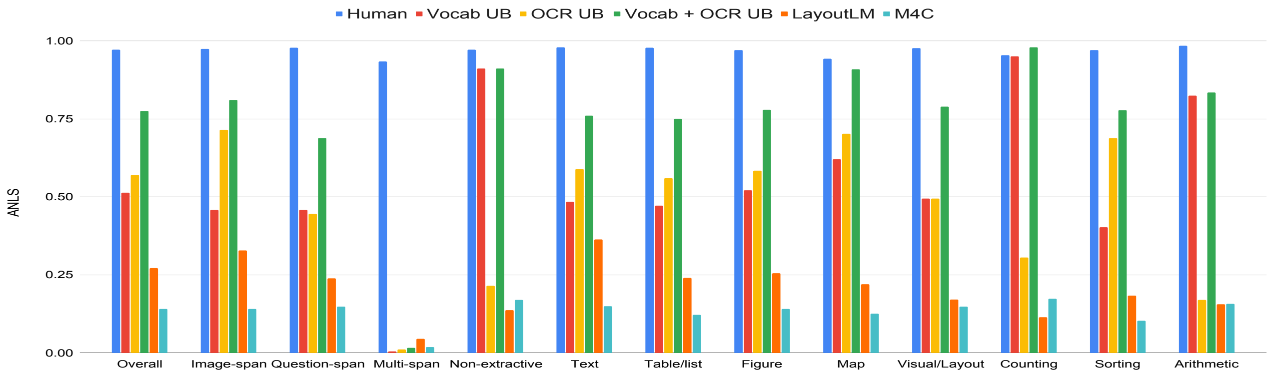
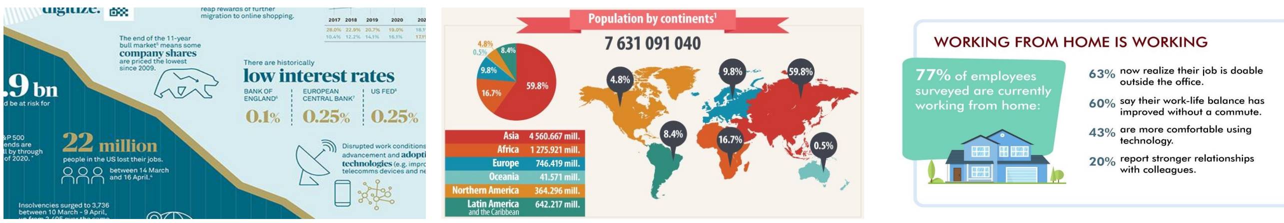


Figure 6: Performance of baselines and upper bounds for different QA types.



What is the interest rates of European Central Bank and US FED?				Which is the least populated continent in the world?				What percentage of workers are not working from home?			
LayoutLM:	0.25%	M4C:	0.1%	LayoutLM:	EU	M4C:	Oceania	LayoutLM:	77%	M4C:	66%
Human:	0.25%	GT:	**0.25%**	Human:	Oceania	GT:	**Oceania**	Human:	23%	GT:	**23%**

Figure 7: **Qualitative Results** For the left most question, evidence is Table/List and the LayoutLM gets it right. In case of the second question where evidence is a Table/List and Sorting is involved, M4C answers correctly. In case of the last question that requires subtraction of 77 from 100 neither M4C, nor LayoutLM gets the answer correct. For better visualization, images we show here are relevant regions cropped from the original infographics. More qualitative examples showing images in original size are given in the supplementary material.

plies that token embeddings of the OCR tokens are good enough and the additional information from visual features of the tokens contribute little to the performance.

Most of the recent models that employ visio-linguistic pretraining of BERT-like architectures [27, 40] incorporate bottom-up visual features—features of objects detected on the images—into the model as visual tokens. We follow the approach in VisualBERT [27], where visual tokens are concatenated after the input stream of text tokens. Each visual token is represented by a dummy text token [OBJ], a separate segment, 1D and 2D positions and the ROI pooled visual feature of the object's region. But in our experiments, the addition of visual tokens did not give us results any better than the model without visual tokens. Hence we do not show this setting in illustration of our model architecture or in the results table. We believe the visual tokens we use impart little information since the object detectors we use— a detector trained for detecting objects on natural scene images and another for document layout analysis— are not suitable for infographics. This is evident from Table 3. Both the detectors detect only a few instances of objects on infographics.

In Figure 6, the performance of our trained baselines on the test split is compared against the upper bounds and human performance. The M4C and LayoutLM models used for this comparison are the variants that give best ANLS on the test data. Finally a few qualitative results from our experiments are shown in Figure 7.

6. Conclusion

We introduce the InfographicVQA dataset and the task of VQA on infographics. Results using the baseline models suggest that existing models designed for multimodal QA or VQA perform poorly on the new dataset. We believe our work will inspire research towards understanding images with a complex interplay of layout, graphical elements and embedded text.

Acknowledgements

This work is supported by MeitY, Government of India, the CERCA Programme / Generalitat de Catalunya and project PID2020-116298GB-I0.

References

[1] Amazon Textract. `https://aws.amazon.com/textract/`. Accessed: 2021-08-16.

[2] Huggingface's Models. `https://huggingface.co/models`. Accessed: 2021-08-16.

[3] stacked attention networks for image question answering.

[4] 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.

[5] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[6] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. Docformer: End-to-end transformer for document understanding, 2021.

[7] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marcal Rusinol, Ernest Valveny, C.V. Jawahar, and Dimosthenis Karatzas. Scene Text Visual Question Answering. In *ICCV*, 2019.

[8] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What Makes a Visualization Memorable? *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2306–2315, 2013.

[9] Z. Bylinskii, Sami Alsheikh, Spandan Madan, A. Recasens, Kimberli Zhong, H. Pfister, F. Durand, and A. Oliva. Understanding infographics through textual and visual tag prediction. *ArXiv*, abs/1709.09215, 2017.

[10] R. Chaudhry, S. Shekhar, U. Gupta, P. Maneriker, P. Bansal, and A. Joshi. Leaf-qa: Locate, encode attend for figure question answering. In *WACV*, 2020.

[11] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: learning universal image-text representations. *CoRR*, abs/1909.11740, 2019.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *ACL*, 2019.

[13] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *NAACL-HLT*, 2019.

[14] Łukasz Garncarek, Rafał Powalski, Tomasz Stanisławek, Bartosz Topolski, Piotr Halama, and Filip Graliński. Lambert: Layout-aware language modeling using bert for information extraction. *arXiv preprint arXiv:2002.08087*, 2020.

[15] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *CVPR*, 2017.

[16] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *CVPR*, 2018.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *CVPR*, 2017.

[18] Himanshu. Detectron2 for document layout analysis. `https://github.com/hpanwar08/detectron2.git`, 2020.

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

[20] Drew A. Hudson and Christopher D. Manning. GQA: a new dataset for compositional question answering over real-world images. *CoRR*, abs/1902.09506, 2019.

[21] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. DVQA: Understanding Data Visualizations via Question Answering. In *CVPR*, 2018.

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

[23] Vahid Kazemi and Ali Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, 2017.

[24] 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.

[25] 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.

[26] Nathan Landman. Towards abstractive captioning of infographics. Master's thesis, Massachusetts Institute of Technology, Massachusetts Institute of Technology, 2018.

[27] Liunian Harold Li, Mark Yatskar, Da Yin, C. Hsieh, and Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language. *ArXiv*, abs/1908.03557, 2019.

[28] Yulin Li, Yuxi Qian, Yuchen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding. Structext: Structured text understanding with multi-modal transformers, 2021.

[29] Min Lu, Chufeng Wang, Joel Lanir, Nanxuan Zhao, Hanspeter Pfister, Daniel Cohen-Or, and Hui Huang. Exploring Visual Information Flows in Infographics. In *ACM CHI*, 2020.

[30] Spandan Madan, Zoya Bylinskii, Matthew Tancik, Adrià Recasens, Kimberli Zhong, Sami Alsheikh, Hanspeter Pfister, Aude Oliva, and Fredo Durand. Synthetically trained icon proposals for parsing and summarizing infographics. *arXiv preprint arXiv:1807.10441*, 2018.

[31] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A Dataset for VQA on Document Images. In *WACV*, 2020.

[32] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. OCR-VQA: Visual question answering by reading text in images. In *ICDAR*, 2019.

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019.

[34] Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. Going full-tilt boogie on document understanding with text-image-layout transformer. *arXiv preprint arXiv:2102.09550*, 2021.

[35] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*. 2015.

[37] 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.

[38] Amanpreet Singh, Vedanuj Goswami, and D. Parikh. Are we pretraining it right? Digging deeper into visio-linguistic pretraining. *ArXiv*, abs/2004.08744, 2020.

[39] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards VQA Models That Can Read. In *CVPR*, 2019.

[40] Weijie Su, X. Zhu, Y. Cao, B. Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *ArXiv*, abs/1908.08530, 2020.

[41] Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. Visualmrc: Machine reading comprehension on document images. *arXiv preprint arXiv:2101.11272*, 2021.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*. 2017.

[43] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[44] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[45] Y. Wu, M. Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, M. Krikun, Yuan Cao, Q. Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, Taku Kudo, H. Kazawa, K. Stevens, G. Kurian, Nishant Patil, W. Wang, C. Young, J. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, G. S. Corrado, Macduff Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144, 2016.

[46] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. *ACM SIGKDD*, Jul 2020.

[47] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020.

[48] Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. RecipeQA: A Challenge Dataset for Multimodal Comprehension of Cooking Recipes. In *EMNLP*, 2018.

[49] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. TAP: Text-Aware Pre-training for Text-VQA and Text-Caption, 2020.

[50] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. PubLayNet: largest dataset ever for document layout analysis. In *ICDAR*, 2019.