Supplementary Materials

A. Detailed Results Analysis

A.1. Within Model Class Analysis

A.1.1 Encoder vs. Decoder

A key difference between encoder-only and (encoder-) decoder-based models is the ability to generate answers beyond the explicit document textual content. This is clearly reflected in the results for BigBird, Longformer, BERT, and LayoutLMv3, which score $<$ 10 ANLS% on abstractive questions, whereas they have just average scores for extractive questions. On **DUDE**, we can claim that a generative model is necessary given all considered question types.

Quite remarkably, while the human baseline demonstrates that humans find abstractive questions (ANLS ±82%) easier than extractive questions (ANLS ±68%), the reverse is true for all machine baselines. A potential confounder for these results could be the difference in output formatting for extractive vs. abstractive answers, which is hard to take into account with ANLS evaluation.

A.1.2 Incorporating Layout & Vision

When comparing T5 with and without 2D position embeddings on the diagnostic categories, we find the highest improvements on 'evidence table or list', 'complexity simple', and 'evidence plain'.

Our study with the proposed baselines shows that questions requiring visual evidence to be answered are an important future challenge for the vision community. To get further insights into models' performance on these questions, we calculate a weighted average of ANLS over visual categories. This reveals that GPT3 (4-shot) and T5-2d-large-8K obtain a tied score of (ANLS=37%), even though they only have access to the text. The human performance, on the other hand, is close to double (ANLS=72%), thus showing the need for better integration of the visual modality in DU models.

A.1.3 Toward Long Document Processing

DUDE clearly requires methods that can process long sequences, as evidenced by its average document length of 1832 ± 2545 tokens. This is particularly evident when comparing standard NLP QA methods like BERT-concat, which underperforms Longformer [6] and BigBird [103], despite being the *large* version. Experiments with T5 and T5-2D further support this claim, as extending the sequence length from 512 to 8192 leads to a $\sim$ 5% ANLS improvement.

The exception is HiVT5 [90], which performs worse than the rest of the methods. This is due to the authors of HiVT5 performing a pre-training task of text denoising that helped to better model the [PAGE] tokens. This resulted in a better, compressed representation of the relevant information within a document conditioned by a question. Moreover, the authors also did extensive experimentation and found that 10 [PAGE] tokens per page were the best fit for the MP-DocVQA [90] dataset. We used similar hyperparameters, but **DUDE** might require better fine-tuning of [PAGE] tokens since the images are more visually rich with colored graphics and layouts. The hierarchical processing of documents with a meaningful visual component is a promising avenue for future research.

A.1.4 Diagnosis of LLM Results

The reasoning for including these LLMs as baselines stems from our question: "Does advanced text understanding suffice for solving **DUDE**?". Our results for diagnostic categories reveal some strengths and weaknesses of LLMs in the DocVQA task setting.

Strengths GPT3 trumps all other tested models for list-type questions (ANLS=36-40%), which can be explained by the extractive nature of these questions. After 4-shot fine-tuning, ChatGPT (4-shot) is better than all other tested baselines in answering not-answerable questions (ANLS=77.45%). This can partly explain the appeal of this particular GPT checkpoint in recent times. GPT3 (4-shot) outperforms (ANLS=52.51%) other tested baselines on questions from the 'complexity multi-hop' category such as *What city name appears the most often in the timetables?*.

Weaknesses Compared to another (more simple text-only generative baseline, T5-base-512 (ANLS=47%), LLMs perform two times worse on abstractive questions (ANLS=22%). Closer analysis reveals that LLMs (even after 4-shot fine-tuning) predict abstractive questions to be *Not-answerable* in 55% of cases (in reality: 10%). Operations such as arithmetic, counting, and comparisons remain generally elusive skills (<25%ANLS).

Both LLMs we tested scored significantly lower than the human baseline in questions that require visual understanding, with an average ANLS score of 21%. This is understandable because these are text-only models.

While LLMs' zero-shot performance is relatively high, we note that **DUDE** consists of public-license documents from the web, which potentially might have been included in the LLMs' pre-training corpus.

A.2. Assessing Confidence

ECE measures calibration of confidence, whereas AURC assesses both performance and confidence ranking [34] (more detail Appendix B.4). The latter results in an appropriate metric to select the best model in real-world applications, where wrong predictions can yield undesired scenarios, which could be prevented by manually revising low-confidence answers.

Interestingly, T5-base-512 scores better on calibration

Model	ANLS	ECE	AURC
BertQA MPDocVQA Concat	29.8	**13.83**	**43.28**
BertQA MPDocVQA MaxConf	**32.18**	28.93	48.73
BigBird MPDocVQA Concat	**30.67**	**25.07**	**47.2**
BigBird MPDocVQA MaxConf	29.38	50.79	56.81
LayoutLMv3 MPDocVQA Concat	22.61	**13.19**	57.11
LayoutLMv3 MPDocVQA MaxConf	**25.27**	31.31	58.54
Longformer MPDocVQA Concat	**33.45**	**22.21**	45.83
Longformer MPDocVQA MaxConf	28.67	48.6	58.11
T5 MPDocVQA Concat	34.37	**18.97**	47.31
T5 MPDocVQA MaxConf	**37.56**	23.73	**46.69**
T5-base Concat-0	**25.62**	**20.05**	62.25
T5-base MaxConf-0	22.21	39.47	**58.89**

Table 4: Comparison of baselines using Concat or Max Conf strategies.

Hyper-Parameter	T5	T5+2D	HiVT5
Epochs	10	10	10
Warm-up (iterations)	1000	250	1000
Optimizer	Adam, AdamW	Adafactor	Adam
Gradient acc.	False	8	False
Lower case	True	True	True
Max. Seq. Length	512, 8192	512, 8192	20480
Generation (Max. Tokens)	100	100	50
Batch size	3	8	1
Learning rate	1E-04, 2E-04	2E-04	2E-04
Training time (per epoch)	1h, 10h	1.5h, 5h	10h
GPU Hardware	TITAN RTX, A100	A100 (80GB)	TITAN RTX (24GB)

Table 5: Hyperparameters used for fine-tuning `T5`, `T5-2D` and `HiVT5` on **DUDE**. When two values are placed in a single column, they refer to the model's versions with 512 and 8192 input sequence length, respectively.

(ECE=10.82) than T5-2D-large-8K, the baseline with the highest ANLS, yet worse calibration (ECE=14.4). In general, it seems calibration worsens when extending the maximum sequence length, whereas adding 2D position embeddings only positively affects ANLS. From the baselines tested, T5-2D-large-8K achieves the highest AURC.

Another interesting result comes from analyzing the calibration of models evaluated using the *Concat* strategy vs. *Max Conf.* strategy. In the main paper, we reported results for the model with the relative best ANLS. Thanks to our varied set of evaluation metrics, we discover that *Max Conf.* overall results in poor calibration (see Table 4), whereas considering ANLS, there is not always a clear winning strategy. This shows that predicting each page separately and necessarily assuming conditional independence across pages is not a reliable strategy for multipage DocVQA.

B. Baseline Experiments Setup

In this Section, we describe the implementation details[6] for the architectures and inference methods used in our benchmark.

B.1. Hyperparameter Defaults

Refer to Table 5.

B.2. Generative LLM Prompt Fine-tuning

The performance of GPT3.5 models was assessed in two settings: 0-shot and 4-shot. In the 0-shot setting, the prompt included instructions similar to those provided to annotators to teach them how to annotate. In the 4-shot setting, the prompt was enhanced with the content of a single document from the training set along with four questions of different types (extractive, abstractive, list, and not answerable) to better gauge the models' abilities.

Few-shot Prompts

Document:
<content of single training set document>
–
Question: <extractive question>
Answer: <extractive answer>
–
Question: <abstractive question>
Answer: <abstractive answer>
–
Question: <list question>
Answer: <list item> | <list item>
–
Question: <not-answerable question>
Answer: none
–
Document:
<content of evaluated document>
–
Questions and answers pairs to above document:
Answers contains either:
- a span inside of document
- a list of spans inside of document (each span should be separated by "|")
- not exist explicitly as span of document (the answer should be freely generated text)
- question couldn't be answered (the answer should be "none")
Question: <question>
Answer: ______

The 0-shot prompt is analogous to the 4-shot prompt, but the key distinction is that it lacks the first document and the example question-and-answer pairs.

[6] Main framework used: https://github.com/rubenpt91/MP-DocVQA-Framework

For the inference process, we utilized the prompt completion default settings outlined in the OpenAI documentation, with the exception of the temperature parameter, which was lowered to a value of 0.0. This adjustment was made to ensure that the output would be more deterministic and focused, with less emphasis on generating creative variations.

Only after our prompting experiments had been completed, we realized the opportunity to assess confidence estimation using chained prompts (*Please give a confidence between 0 and 1 about how certain you are this is the answer.*) as in [40]. Since we did not save our dialogue states and considered the expenses, we leave this for future work.

B.3. Confidence Estimation

This Subsection details confidence scoring functions for the baselines, as this is not reported in standard practice.

We define *confidence* as the predicted probability of the top-1 prediction, often arising as the largest value from softmax normalization of logits from a final model layer (head).

Encoder-based models will output logits for all possible start and end positions of the answer within the provided context. While the predicted answer of such a span prediction architecture will come from the highest valid (no negative span) combination of the sum of a start and end logit, the predicted answer confidence can be obtained by the following procedure (BS: batch size and S: sequence length):

```
# Standard span prediction forward call
outputs = model(**inputs,
↪  start_positions=start_positions,
↪  end_positions=end_positions)

# Assumes masking all padding and special tokens
↪  after softmax with 0
start = outputs.start_logits.softmax(dim=1)
.unsqueeze(dim=0).unsqueeze(dim=-1) #1 x BS x S x
↪  1
end = outputs.end_logits.softmax(dim=1)
.unsqueeze(dim=0).unsqueeze(dim=1) #1 x BS x 1 x
↪  S

# Compute the probability of each valid (end <
↪  start) start, end pair
candidate_matrix = torch.matmul(start,
↪  end).triu().detach().numpy() # 1 x BS x S x S

# Obtain highest scoring candidate span by
↪  multi-index argmax
flat_probs = candidate_matrix.reshape((1, -1)) #
↪  BS x S*S
batch_idx, start_idx, end_idx =
↪  np.unravel_index(np.argmax(flat_probs, 1),
↪  candidate_matrix.shape)[1:]
batch_answer_confs = candidate_matrix[0,
↪  batch_idx, start_idx, end_idx]
```

Decoder-based models are not restricted to spans and can output an arbitrary, though often controllable, amount of text tokens, indicated as S'. The logits at the final layer take the shape of $BS \times S' \times V$, where V is the tokenizer's vocabulary size (32.1K for T5-base). The following confidence estimation procedure is applied for decoder outputs:

```
# Standard decoder-based greedy forward pass
↪  (without labels)
outputs = model.generate(**input_ids,
↪  output_scores=True,
↪  return_dict_in_generate=True)

# BS x S' x V, dropping EOS token and applying
↪  softmax + argmax per token
batch_logits = torch.stack(outputs.scores,
↪  dim=1)[:, :-1, :]
decoder_outputs_confs =
↪  torch.amax(batch_logits.softmax(-1), 2)

# Remove padding from batching decoder output of
↪  variable sizes
decoder_outputs_confs_masked = torch.where(
    outputs.sequences[:, 1:-1] > 0,
    decoder_outputs_confs,
    torch.ones_like(decoder_outputs_confs))

# Multiply probability over tokens
batch_answer_confs =
↪  decoder_outputs_confs_masked.prod(1)
```

B.4. Evaluation metrics

All metric implementations (ANLS, ECE, AURC) are made available as a standalone repository. Additionally, we provide an online service where researchers can evaluate their methods against a blind (questions-only) test dataset. Below, we expound on the implementation details of the metrics and motivate design choices.

B.4.1 ANLS

Average Normalized Levenshtein Similarity (ANLS) is a metric introduced in [8], which was then extended [89] to support *lists* and be invariant to the order of provided answers. We adapt the underlying Levenshtein Distance metric [45] to support *not-answerable* questions, $\mathrm{NA}(G) = \mathbb{I}[\mathrm{type}(G) = \text{ not-answerable }]$ (see Equation (1)).

Consider for simplicity, the evaluation of a single non-list ground truth answer G and prediction $\hat{P}$, each with string lengths $|G|$ and $|\hat{P}|$, respectively.

$$\mathrm{LD}(G,\hat{P}) = \begin{cases} 1 \quad \text{if } \mathrm{NA}(G) \wedge |\hat{P}| > 0, \\ 0 \quad \text{if } \mathrm{NA}(G) \wedge |\hat{P}| = 0, \\ |G| \quad \text{if } |\hat{P}| = 0, \\ \mathrm{LD}(\mathrm{tail}(G), \mathrm{tail}(\hat{P})) \quad \text{if } G[0] = \hat{P}[0], \\ 1 + \min \begin{cases} \mathrm{LD}(\mathrm{tail}(G), \hat{P}) \\ \mathrm{LD}(G, \mathrm{tail}(\hat{P})) \\ \mathrm{LD}(\mathrm{tail}(G), \mathrm{tail}(\hat{P})) \end{cases} \text{, otherwise} \end{cases} \tag{1}$$

The normalized similarity metric is then defined as

$$\mathrm{NLS}(G, \hat{P}) = \frac{1 - \mathrm{LD}(G, \hat{P})}{\max(1, |G|, |\hat{P}|)}.$$

Given multiple ground truth answer variants $G = \{g_1, g_2, ...\}$ and a predicted answer for $\hat{P}_{Q_i}$ for each question Q in the test set of size N, we define the complete metric as follows:

$$\mathrm{ANLS} = \frac{1}{N} \sum_{i=0}^{N} \left(\max_{g \in G_i} s\left(g, \hat{P}_{Q_i}\right) \right) \tag{2}$$

$$s\left(g, \hat{P}_{Q_i}\right) = \begin{cases} \mathrm{NLS}\left(g, \hat{P}_{Q_i}\right) & \text{if } \mathrm{NLS}\left(g, \hat{P}_{Q_i}\right) \geqslant \tau \\ 0 & \text{if } \mathrm{NLS}\left(g, \hat{P}_{Q_i}\right) < \tau \end{cases}, \tag{3}$$

where we follow prior literature [8, 89] in setting the threshold $\tau = 0.5$.

In the case of a *list*-type question, Hungarian matching is performed following [89] according to NLS between each ground truth answer part and each prediction answer part.

While ANLS can account for shortcomings of OCR and formatting issues, evaluation of generated text is notoriously complex [74] and requires more research.

B.4.2 ECE

Expected Calibration Error (ECE) is a default metric to evaluate top-1 prediction miscalibration. It measures the $\mathcal{L}_p$ norm difference between a model's posterior and the true likelihood of being correct, as formally defined below:

$$ECE_p(f)^p = \mathbb{E}_{(X,Y)} \left[\| \mathbb{E}[Y = \hat{y} \mid f(X) = \hat{p}] - f(X) \|_p^p \right],$$

where $\hat{y} = \arg\max_{y'} [f(X)]'_y$ is a class prediction with associated posterior probability $\hat{p} = \max_{y'} [f(X)]'_y$.

In our setting, the exact accuracy condition $\mathbb{I}[Y = \hat{y}]$ is replaced by $\mathbb{I}[\mathrm{ANLS}(y, \hat{y}) > \tau]$. Prior work [62] already introduced the strategy of thresholding continuous quality scores (in the case of IOU larger than τ) in order to be able to estimate ECE.

In practice, ECE is implemented as a histogram binning estimator that discretizes predicted probabilities into ranges of possible values (bins) for which conditional expectation can be estimated. In order to minimize the drawbacks inherited from histogram binning, as suggested by the literature [66, 94, 41, 79], we apply an equal-mass binning scheme with 100 bins (close to $\sqrt{N}$).

B.4.3 AURC

Area-Under-Risk-Coverage-Curve (AURC) [24, 34] measures the possible trade-offs between coverage (proportion of test set%) and risk (error % under given coverage). It assumes predictions to come with a confidence estimate. The curve can be obtained by sorting all confidence estimates and evaluating risk from high to low, together with their respective correctness (typically based on exact match).

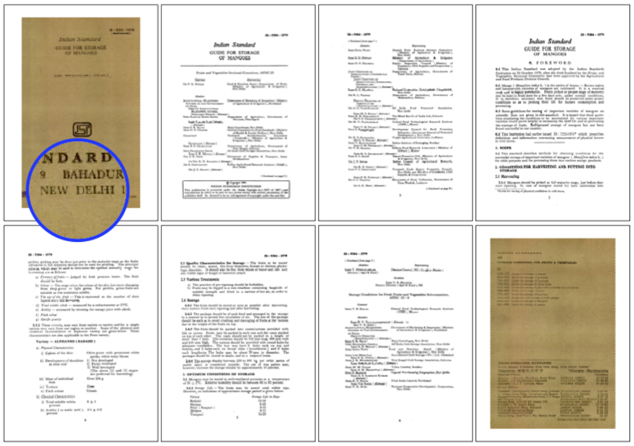
Similar to ECE as defined above, we apply ANLS thresholding instead. Formulated this way, the best possible AURC is constrained by the model's test error (1-ANLS) and the number of test instances. We have extended the very detailed implementation of [34], to which we refer for further information. On a final note, AURC might be more sensible for evaluating highly-accurate settings (e.g., 90% accuracy), where risk can be better controlled (as it is typically a business decision to decide tolerance to mistakes).

C. Qualitative Examples

As is customary, we provide some interesting, hand-picked test set examples with predictions from some of the baselines in our study.

Low complexity. *Where the document has been printed?*

Simple, extractive question, plain-text evidence.



Source	Answer	ANLS	Conf.
Ground truth	*New Delhi, India*		
Human	*India*	0.0	—
T5	*IS : 9304 - 1979*	0.0	0.56
ChatGPT	*The document does not mention where it has been printed.*	0.0	—
GPT3	*Bela Pack n Print. New Delhi, India*	0.0	—
T5-2D	*New Delhi, India*	1.0	0.09
HiVT5	*Page 1*	0.0	0.18
Longformer	new delhi, india	1.0	0.72

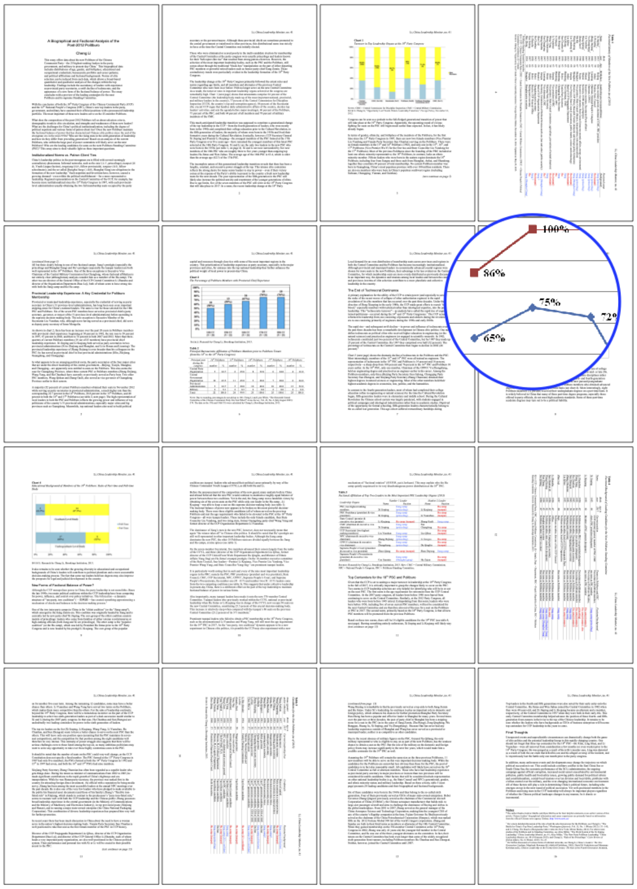
High complexity. *Is there any redacted section on the document?* Abstractive question that requires knowledge about possible document elements.

Source	Answer	ANLS	Conf.
Ground truth	*No*		
Human	*No*	1.0	—
T5	*yes*	0.0	0.17
ChatGPT	*[Not-answerable]*	0.0	—
GPT3	*[Not-answerable]*	0.0	—
T5-2D	*No*	1.0	0.43
HiVT5	*Yes*	0.0	0.55
LayoutLMv3	*approved for release*	0.0	0.01

Requires arithmetic. *What is the difference between how much Operator II and Operator III makes per hour?* The question requires table comprehension, determining relevant values, dividing extracted integers, and correcting the subject-verb agreement.

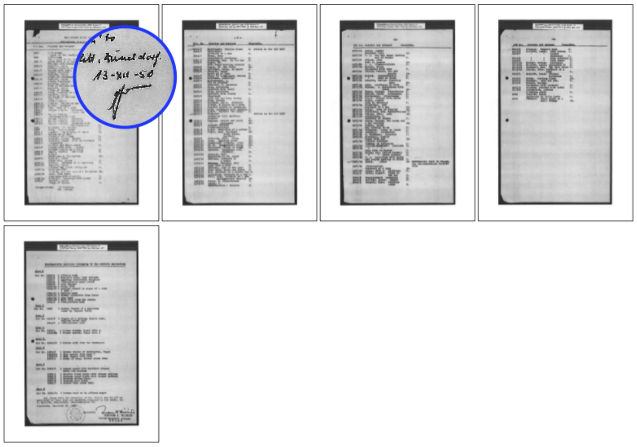
Source	Answer	ANLS	Conf.
Ground truth	*$5*		
Human	*$5*	1.0	—
T5	*200*	0.0	0.28
ChatGPT	*$5 per hour.*	0.0	—
GPT3	*Operator II ($17/hr) | Operator III ($22/hr)*	0.0	—
T5-2D	*[Not-answerable]*	0.0	0.31
HiVT5	*[Not-answerable]*	0.0	0.15

Visual evidence (chart). *What is the maximum percentage of the blue graph line on page 8?* A highly demanding question that requires simultaneous competency of visual comprehension (locating chart and line color), navigating through layout (determining adequate page), and numerical comparison (deciding on the highest value).



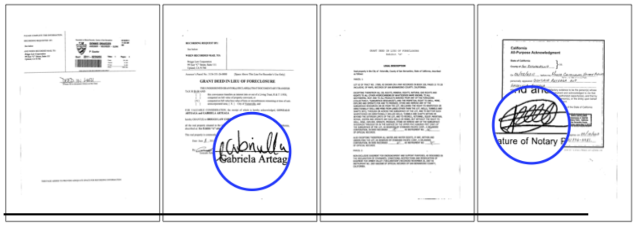
Source	Answer	ANLS	Conf.
Ground truth	*75%*		
Human	*75*	0.7	—
T5	*76*	0.0	0.25
ChatGPT	*[Not-answerable]*	0.0	—
GPT3	*76%*	0.7	—
T5-2D	*32.0*	0.0	0.00
HiVT5	*45%*	0.7	0.05
BigBird	*32*	0.0	0.47
LayoutLMv3	*80%*	0.0	0.15

Visual evidence (handwriting). *What is the handwritten date on page 1?* The question requires visual comprehension (recognition of handwriting) and layout navigation (determining the adequate page).



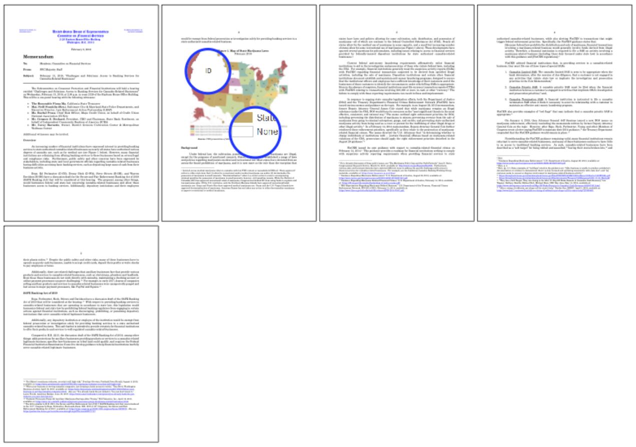
Source	Answer	ANLS	Conf.
Ground truth	*13-XII-50*		
Human	*13-XII-50*	1.0	—
T5	*1977-01-01*	0.0	0.24
ChatGPT	*[Not-answerable]*	0.0	—
GPT3	*15 December 1950*	0.0	—
T5-2D	*1950-12-15*	0.0	0.24
HiVT5	*1977-07-01*	0.0	0.11
BERTQA	*2006 / 1*	0.0	0.5

Requires counting. *How many pages have a signature?* The question requires visual comprehension (recognition of signature), knowledge about layout, and counting.



Source	Answer	ANLS	Conf.
Ground truth	2		
Human	2	1.0	—
T5	*1*	0.0	0.01
ChatGPT	*4*	0.0	—
GPT3	*[Not-answerable]*	0.0	—
T5-2D	*4*	0.0	0.69
HiVT5	*4*	0.0	0.41

Visual evidence (map), multi-hop. *Which states don't have any marijuana laws?* The multi-hop question requires visually comprehending the map and linking knowledge from its legend with depicted regions.



Source	Answer	ANLS	Conf.
Ground truth	*ID ǀ SD ǀ KS*		
Human	*ID ǀ SD ǀ KS*	1.0	—
T5	*WA ME MT ND MN OR VT ID NH SD WI NY MA MI*	0.0	0.28
ChatGPT	*[Not-answerable]*	0.0	—-
GPT3	*American Samoa*	0.0	—-
T5-2D	*i*	0.0	0.03
HiVT5	-	0.0	0.02

D. Additional Dataset Statistics

D.1. Answer Types

Figure 6 shows that there are barely any correlations between question type and answer type, except for the most expected ones (e.g. 'None' answers and 'Not answerable' questions), by means of Cramer's V coefficient. For instance, date and duration types of answers are equally likely for both extractive and abstractive questions.

Figure 7 shows the answer type distribution per question type in **DUDE**, followed by a comparison to answer type distributions in related DocVQA datasets (Figure 8).

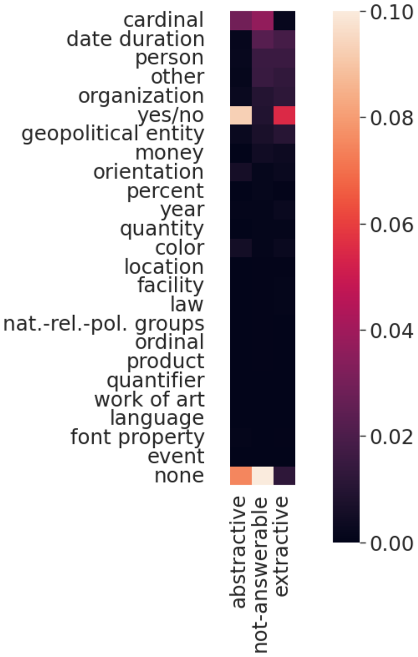


Figure 6: Answer types correlation heatmap. Results obtained with Cramer's V coefficient. Note that values on the scale are below 0.1, suggesting a lack of correlation.

D.2. Dataset Diversity

Similar to the text-based comparison, Figure 9 visualizes the diversity of the visual embeddings of all documents' first pages in **DUDE**, relative to those from other DocVQA datasets.

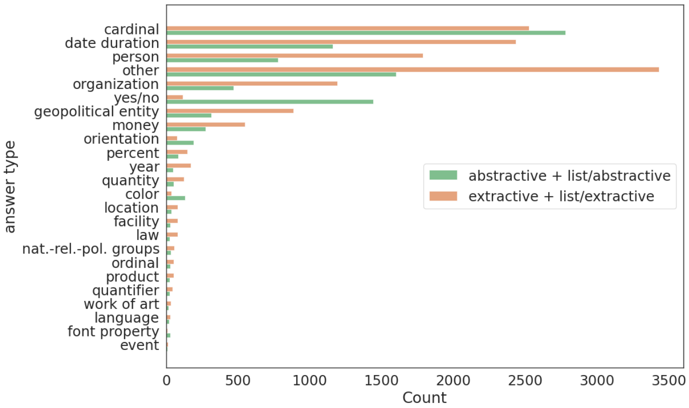


Figure 7: Answer type distribution per question type in **DUDE**.

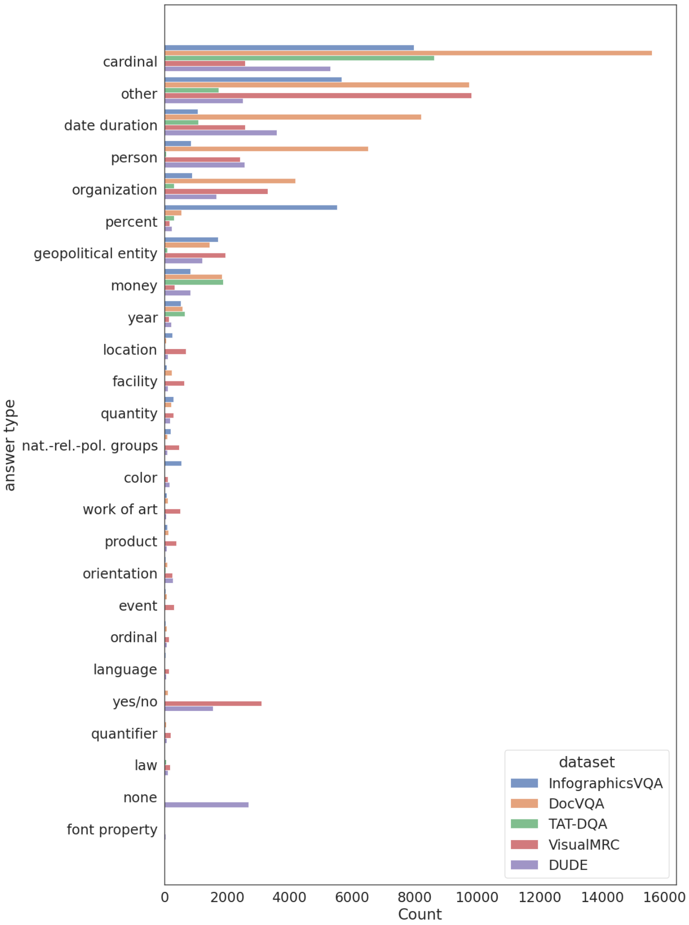


Figure 8: Answer type distribution per dataset, sorted in descending order of total answer type occurrences. We have found: 13 answer types in TAT-DQA; 20 answer types in InfographicsVQA and SP-DocVQA, 23 answer types in VisualMRC, and 24 answer types in **DUDE**

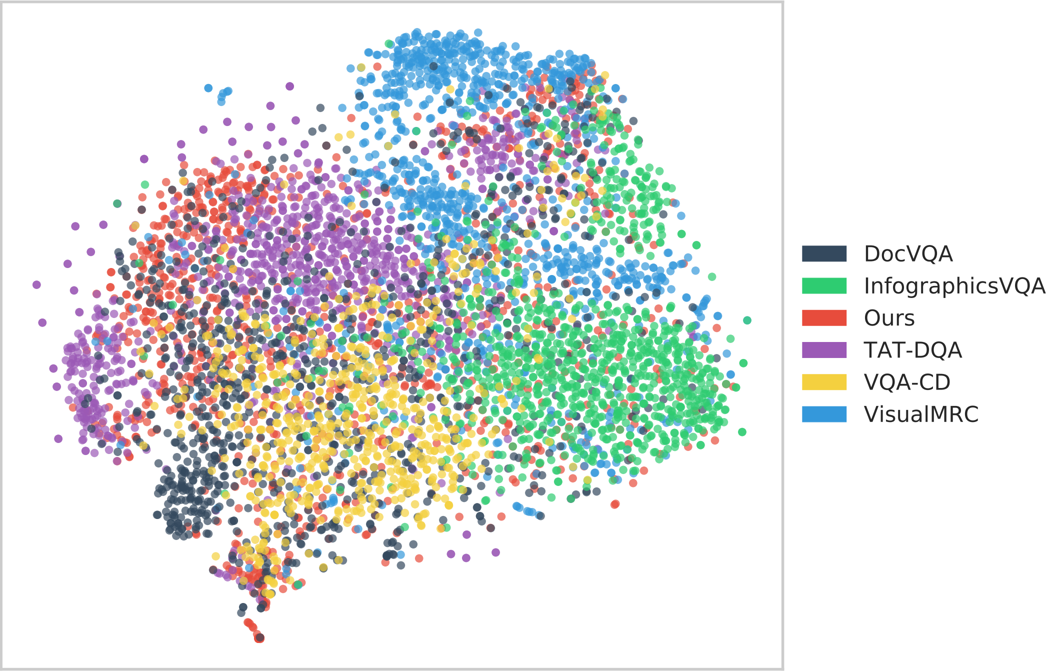


Figure 9: Visualization of document image similarities between samples from different datasets (t-SNE over ResNet101 features of 1k documents, first pages only).