ScreenQA: Large-Scale Question-Answer Pairs Over Mobile App Screenshots

Yu-Chung Hsiao∗† Fedir Zubach∗ Gilles Baechler Victor Cărîbune Jason Lin
Maria Wang Srinivas Sunkara Yun Zhu Jindong Chen

Google DeepMind
jdchen@google.com

Abstract

We present a new benchmark and dataset, ScreenQA, for screen content understanding via question answering. The existing screen datasets are focused either on structure and component-level understanding, or on a much higher-level composite task such as navigation and task completion. We attempt to bridge the gap between these two by annotating 86K question-answer pairs over the RICO dataset in hope to benchmark the screen reading comprehension capacity. This work is also the first to annotate answers for different application scenarios, including both full sentences and short forms, as well as supporting UI contents on screen and their bounding boxes. With the rich annotation, we discuss and define the evaluation metrics of the benchmark, show applications of the dataset, and provide a few baselines using closed and open source models.

1 Introduction

Recent advent of machine learning, especially Visual Large Language Models (VLMs), has motivated a long list of applications that are based on mobile screens. To name a few, a personal agent for users to use mobile device hands-free or eyes-free, code generation from UI design mock-ups, adaptation of device UI, automatic ads generation, testing and criticising mobile apps, all require advanced technology in understanding or annotation of the mobile screens.

Mobile app screenshots have been analyzed using machine learning from multiple aspects. These analyses range from pixel level understanding, e.g., layout structural analyses, UI issue detection and correction (Li et al., 2022), to UI element semantics, e.g., icon recognition, button action prediction (Sunkara et al., 2022), to even higher-level functional analyses such as accessibility support (Li et al., 2020b), screen description (Wang et al., 2021), and screen type classification (Deka et al., 2017). Comparatively, the content understanding aspect is understudied. Examples of content include star ratings from restaurant reviews, flight status, and messages from chats. Having this capacity of understanding is important for two reasons: First, most user activities are information seeking. Second, transactional task (e.g. book a flight) requires precise understanding of the UI content including actionable ones, and their state. In this work, we advocate to use only screenshot as the sole representation of UI screens, as the underneath structure representation could be messy. With this setting, it is important to devise a task to measure quantitatively information detection, extraction and understanding from screenshot images.

∗co-first authors with equal contributions.
†work done while at Google.

Preprint. Under review.
To this end, we annotated the RICO dataset \cite{Deka2017} with 85,984 question-answer pairs, referred to as ScreenQA, and released the dataset in the public domain. In each pair, the answer is to the question regarding a screenshot. Each answer is a 4-tuple, a short answer (e.g. a phrase), a full sentence answer, a list of UI contents supporting the answer, and a list of bounding boxes of the contents. See examples in Figure 1. The components in the answers, individually or in combination, serve different purposes in metrics for tasks that are motivated by real-life applications, for example, short answers for information extraction, long answers for summarization, UI content for textual grounding in entity extraction, and bounding boxes for location grounding. To the best of our knowledge, this is the first large-scale questions answering dataset over mobile app screenshots, and the first one to be publicly available. Inspired by the SQuAD dataset \cite{Rajpurkar2016}, we hope to encourage the community to advance technologies toward better screen content understanding, which will benefit beyond itself and human computer interaction (HCI) community.

The contributions of this work are the following.

1. We create and release ScreenQA, a large-scale question answering dataset and benchmark for mobile screens, as the first of its kind.
2. ScreenQA is also the only dataset with rich annotations, including short answers, full sentence answers, supporting UI content, and their bounding boxes.
3. Based on the rich annotations, we define 4 tasks and corresponding metrics motivated by the need to measure quality in real-life applications.
4. We build 2 baselines with closed and open source models (ScreenAI 5B \cite{Baechler2024}, PaLIGemma \cite{development_contributors2024}), and release open source model checkpoints.

2 Related work

UI graphics are typically designed to be informative and actionable for users. Existing datasets can be categorized by their associated tasks, which we outline below.

**Screen and UI-based Understanding** In contrast to natural images, screen images are composed of structured components generated by a tree-like syntax of view hierarchy (VH) or Document Object Models (DOM). Previous works identify elements, i.e. icon detection \cite{Deka2017}, widget captioning via language \cite{Li2020b}, and referring expression in various classification \cite{Wu2023}, retrieval \cite{Bai2021} and generation \cite{Hong2023} setups. Meanwhile others specialize in challenging variations i.e. aspect ratio, sizes, OCR. In developing web agents, MoTIF \cite{Burns2022} and VisualWebArena \cite{Koh2024} provide interactive app environments for evaluating visually grounded screen agents. Rather than high-level goal-completion, our focus is on content understanding of language heavy UI in the broader context for human interaction.

**Multimodal Question Answering** Question answering tasks can be categorized by 1) open- or closed-domain and capacities to evaluate ranging from reading comprehension, multi-hop reasoning and logic reasoning. ScreenQA is a closed-domain question answering task that expects answers by span (or UI element phrase) selection for screen reading comprehension. As described in Section 2, we instructed the data annotators to avoid multi-hop, mathematical counting, and logic reasoning, in order to focus on the fundamental screen comprehension capacity. With the flexibility of both short and long-form answers, along with grounded bounding boxes in a Question-Answering format, ScreenQA natively aligns with most LLMs' retraining recipe.

**Document Image Understanding** Given Vision-Language Models (VLMs)'s limitations in counting and logical reasoning, recent datasets like InfographicQA \cite{Mathew2022}, ChartQA \cite{Masry2022} are created. Mobile app screenshots contain nearly all possible representation of information, specifically text, blended with icons, symbols, and images through pixels. This makes it similar to the understanding of scanned or photographed documents. DocVQA \cite{Mathew2021} uses an extractive QA format for span/segment extraction. Along with TextVQA and several other domain-specific datasets \cite{Mishra2019, Kahou2017} with applications i.e. for textbooks.

\footnote{ScreenQA dataset is released at \url{https://github.com/google-research-datasets/screen_qa}}
Figure 1: ScreenQA examples. (a) Short Answer: "1,1", Full Answer: "There is 1 like and 1 comment", UI Content: "1","1", Bounding boxes are shown in green.

(Singh et al., 2019), math problems, receipts, we believe that techniques developed for relating the 2D arrangement of text are applicable to screens.

3 Data annotation

We perform several steps to collect the ScreenQA annotations, as depicted in Figure 2. Each step is described below. See Appendix C for collected question-answer data examples.

3.1 Pre-filtering

The pre-filtering stage filters out 1) screenshots from non-English apps (different than “non-English screenshots”, as translation and dictionary apps could cause confusion) and 2) screenshots whose view hierarchies (VHs) are out of sync with the main contents. It is a known issue that in the RICO dataset, some screenshots and their corresponding view hierarchies are not perfectly synchronized: there exists certain time difference between view hierarchy extraction and screenshot capturing (Zang et al., 2021). We remove those screenshots to ensure that all ScreenQA annotations are not subject to such data noises.

Classifying the sync quality is tricky, even for human readers. One may not be able to differentiate between occlusion, ghosting, and the actual out-of-sync. See Figure 3 for examples. Accordingly, we instructed the annotators to focus on the main content area of the screen and make sure the bounding boxes in that area are not corrupted, as this is where most contents of interest and questions come from.

We use 27 annotators to perform this step. Among RICO’s 66K unique screenshots, about 11K screenshots are from non-English apps, and about 13K screenshots have out-of-sync view hierarchies. This out-of-sync number is different from (Li et al., 2020a) because we focus on the main content area. After filtering, we are left with about 51K screenshots from English apps with in-sync VHs.
3.2 Question annotation

For question annotation, we asked the annotators to frame questions given a screenshot as the context. The annotators were expected to compose natural, daily-life questions as if using the app. The composed questions should inquire information that can be directly read from the screen and should not require logical reasoning, counting, calculation, or mathematical comparison, etc. We further required the annotators not to ask questions about any advertisement on the screen.

The annotation UI tool is depicted in Appendix A.1. We ask the annotators to compose up to five questions given a screenshot in the first pass. In the second pass, we ask for up to three questions given a screenshot and the questions previously composed. Each pass involve one annotator for each screenshot and whoever annotated the screenshot before is excluded from being assigned to the same screenshot. This ensures that every screenshot is assigned precisely two annotators to compose questions. We choose this sequential process to avoid tricky deduplication of similar questions, and to encourage annotators to diversify their questions. Note that the same set of annotators are involved in both passes such that each annotator has an opportunity to develop their own question style in the first pass before seeing others’ questions in the second pass. This ensures that we still have certain numbers of question styles in the dataset before they converge to each other in repeated passes.

We again involved the 27 annotators. The first pass of question annotation generate 46K questions, and the second pass add an additional 36K questions, resulting in 82K questions in total. Around 15K screenshots are left with no question, due to a lack of interesting content.

3.3 Answer annotation

We use the total 82K questions with 35K distinct screenshots from the previous two-pass question annotation step to further annotate the corresponding answers. The annotator who composed the question is excluded from annotating their own answer to avoid potential biases. Our answer annotation UI tool is shown in Appendix A.2.

Given an example, which contains a screenshot and a question, the annotators are tasked to

1. Fix any grammatical errors or typos of the given question without altering its intention.
2. Answer the question, based on the context of the given screenshot, by 1) selecting bounding boxes from the underlying view hierarchy leaf nodes that contain the relevant answers, or drawing bounding boxes if no suitable leaf nodes can be used, and 2) ranking the answers in descending order of relevance if applicable, or by the common reading order.
3. Additionally also provide a full-sentence answer to the question.
4. Consider two exceptions: 1) The question may be incomprehensible or 2) the screenshot does not contain the answer to the question, due to the questioner’s lack of understanding.
of the app. Then the example should be marked as “invalid question” and “not answerable from the screenshot”, respectively.

5. One answer is annotated for the train split, and three for the validation and the test splits. This is to improve the evaluation quality. More details on the data split details are provided in Section 4.1.

The “invalid question” annotations are then filtered out, and the questions that have no other answer annotations are excluded from the overall ScreenQA dataset, as they are considered incorrectly annotated during the question annotation phase.

3.4 Not-answerable question annotations

The questions marked as “not answerable from the screenshot” represent a special category of questions that check model overtriggering (attempting to answer those which are not supposed to be answered). Being able to come to a conclusion that the answer is not present on the screen is an important aspect of screen understanding. Note that it is possible that one annotator considered a question to be not answerable, and another provided an answer to that same question.

As described in Section 3.2, the first two passes of question annotations aimed to compose questions that can be answered from the screen, so as expected, the fraction of not answerable questions was small. We then had a third pass of question annotation to raise this fraction to nearly 10%, see Figure 4. For this, we used nearly 5K screenshots selected randomly from those where there were no such questions yet. In this pass, we asked annotators for exactly one additional question per screenshot that had some relation to the information there, but could not be answered. See example in Figure 1c. Answer annotation was not used for these 5K questions.

3.5 Short answers generation

One may argue that the exact UI elements containing the answer to a user’s question are not directly utilizable by the user, as it is not always straightforward to convert it to the answer, which is the only important thing in many such scenarios. With that intention an alternative answer information was produced for each question: a short answer.

There are many ways to represent the same information. For example, “25.01.2023”, “25th of January 2023” and “January 25, 2023” represent the same date, and the model should not be penalized for choosing one over the others. To allow this flexibility, multiple answers were produced per question, covering various representations of the same factual answer. The ground truth for each question from this ScreenQA dataset is therefore a list of possible short answers.

A version of the PaLM 2 model (Anil et al., 2023) was used to generate this list of short answers in a few-shot setting. Textual information of the ScreenQA dataset (question, list of UI element descriptions and full-sentence answer) was used as input. See Appendix B for details about the prompts used. The generated lists were then verified by simple heuristics and eyeballing of randomly selected samples.
Table 1: Top ($\geq 1.0\%$) question category distribution and examples. Please see Appendix D for more categories.

| Category                        | %   | Examples                                                                 |
|---------------------------------|-----|--------------------------------------------------------------------------|
| UI selection & config           | 18.1| Which option is selected? What is the selected ringtone?                 |
| Quantity number                 | 11.7| How many unread messages? How many pictures are there in Western Europe? |
| App name                        | 10.4| What is the name of the application? What is the app name?               |
| Date time                       | 9.4 | When was “Heal the Living” released? When is happy hour?                 |
| Price                           | 3.4 | How much is the gift bonus in 3rd place? What is the price?              |
| Name of item                    | 3.3 | What is the name of the drug? What is the name of chef?                  |
| User name                       | 2.8 | What is the name of the user? What is the username on telegram?          |
| Duration                        | 2.5 | What is the duration of video? How long is the song?                     |
| Enum. of avail. options         | 2.5 | Which social media options are given there? What are the options available for logging in? |
| Address and direction           | 2.4 | What is the current location? What is the service zip code?              |
| Email address                   | 2.4 | What is an email address? What is customer service email?                |
| Person’s name                   | 2.1 | Who sang the song? What is the last name?                                |
| Signup/login                    | 1.6 | Which application can be used to sign up / login? What are the alternative choices for signing up? |
| Version information             | 1.6 | What is the version number? What is the new feature in version v3.1.3?  |
| Weather                         | 1.5 | What is the range of temperature shown on Sunday? What is the weather forecast for Sunday? |
| Score & value                   | 1.4 | What is height/weight of the person? What is the score?                  |
| Yes/No                          | 1.1 | Is there any travel plans? Is there any favorite?                        |
| Phone number                    | 1.0 | What is the phone number? What is the prefix for the international mobile number? |
| Others                          | 20.8| What’s the average speed? What is the user’s middle initial              |

(a) Number of composed questions per screenshot.  (b) Number of bounding boxes used to answer the question.

Figure 6: Histograms for number of composed questions and number of bounding boxes in answers. a) The three question annotation passes were capped at five, three and one questions, respectively, resulting in a maximum of nine questions in total. b) The cases with no answer or a single bounding box amount for 91-92% of the answers, they have been removed from the chart in favor of more clarity on the long tail. Answers with 10 or more bounding boxes amount for less than 0.15%.

4 Dataset analysis

4.1 Dataset statistics

The ScreenQA dataset contains 85,984 questions from 35,352 distinct screenshots. It is split into train, validation and test sets in an approximately 80-10-10 ratio, see Table 5. Note that questions related to same screenshot belong to the same split.

4.2 Question analysis

Among the 86K collected questions, there are 47.5K unique questions. It is natural and valid to ask the same common questions over various screenshots, for example, “Which option is selected on the screen?” and “What is the email address?”. Some screenshots receive more questions because they usually contain more information to be asked about. Yet, the histogram still exhibits a reasonable exponential decay with a mild slope, as depicted in Figure 6a.

To further understand what questions have been asked, we categorize the questions using regular expressions based on a list of empirically determined question categories. The categories are meant
to provide a rough overview of the question annotations and by no means to provide a precise categorization. The distribution and examples by these categories are shown in Table 1. Note that the questions were not composed at the annotators’ full discretion: They are conditioned on the given screenshots. That is to say, the distribution is implicitly influenced by the RICO crawling process. For example, as RICO crawled screen traces from freshly installed apps and did not login an account, a noticeable number of the screen traces end at a login page. This in turn translates to a higher percentage of questions asked about app names, email addresses, permissions to login, etc.

4.3 Answer analysis

We analyze the answer annotations in two aspects: 1) How often do we need more than one bounding box and its text to answer the question, and 2) How often do human annotators find the view hierarchy useful to provide a minimal answer to the question.

Figure 6b illustrates the histogram of number of bounding boxes used in each answer. About 84% of answers contain a single bounding box. Among these single-bounding-box answers, 51% uses a VH leaf node directly, while 49% uses a manually drawn bounding box. If we consider all answers together, 51% answers are only based on VH leaf nodes, while 48% uses only manually drawn bounding boxes. Interestingly, for 0.8% of the answers, the human annotators used a mixture of VH leaf nodes and manually drawn bounding boxes. These cases usually happen 1) when the answer is an enumeration of “inhomogeneous” options that are organized differently on the screen, such as using email vs. other APIs to login, and 2) when an answer needs multiple parts to be complete, such as a date consisting of year, month, and day scattered on the calendar UI, and a temperature or a measurement requiring a number followed by the corresponding unit. These parts may not be displayed in the same way, resulting in a lack of useful VH leaf nodes for some of the parts.

Human raters preferred to draw the bounding boxes in about half of the cases; this reflects that the view hierarchy might not necessarily be a very reliable input for ScreenQA.

5 Applications: Tasks and Metrics

We design the data collection guidelines considering several real world applications. In this section, we define metrics accordingly for training and evaluation of models.

Because we collect data from multiple raters for each question in validation and test splits, the metrics accommodate multiple ground truths. We compute an average of the max metric value over all ground truth variants for a given question as \( \text{avg}(\text{metric}) = \frac{1}{N} \sum_{i=1}^{N} \max_{j}[\text{metric}(A_i, A_{i,j}^g)] \), where \( N \) is the number of questions, \( A_i \) is the predicted answer for \( i \)-th question, and \( A_{i,j}^g \) is the \( j \)-th ground truth for \( i \)-th question.

**ScreenQA: Short (SQA-S)**  Given a screenshot and a question, output a short (concise) answer to this question using the information presented on the screen. If the screenshot doesn’t contain the answer, output “<no answer>”.

We highlight this task as one of the key capabilities our dataset enables.

Since there are many ways to represent the same information in text, we produced a list of plausible short answers to be used as ground truth here (see Section 3.5). The two metrics we propose for this task are Exact Match (EM)—to verify answers composed of shorter answers—and F1-Score—to handle acceptable modifications in longer answers, e.g. permutations or rephrasing of quoted content. We apply SQuAD (Rajpurkar et al., 2016b) pre-processing before computing averaged metrics.

**ScreenQA: Long (SQA-L)**  Given a screenshot and a question, output a long (full-sentence) answer to this question using the information presented on the screen. If the screenshot doesn’t contain the answer, output “<no answer>”.

The role of this task is to enable models to output fluent answers that can be directly conveyed to a human, e.g. by an Assistant. Oftentimes, the task resembles a summary of the elements that constitute the answer to the given question. Summary evaluation metrics frequently capture these aspects and therefore we recommend using ROUGE-{1,2,L} (Lin, 2004).

7
ScreenQA: UI Content (SQA-UIC)  Given a screenshot and a question, output a list of UI elements that contain the answer, where each element is represented by its text representation. If the screenshot doesn’t contain the answer, output an empty list. Section 3 describes what UI elements correspond to a question, in which order they should be provided, and what is considered as contents of an UI element.

With the exception of some icons having pre-defined textual descriptions, in most cases this content is text within the UI element, which resembles that of OCR systems [Qin et al., 2019]. The difference, however, is that the output cannot be treated as a continuous sequence of symbols or words. It should be evaluated as a list.

Given elements are lists, we make use of Exact Match and F1-score as metrics. The text present in screenshots is extracted more easily than in arbitrary images and therefore we perform element-wise matching without additional pre-processing.

ScreenQA: UI Content with Bounding Boxes (SQA-UIC-BB)  Given a screenshot and a question, output a list of UI elements that contain the answer, where each element is represented by its bounding box and text representation.

We consider this an extension to the previous SQA-UIC. The exact localization of relevant UI elements allows highlighting the UI elements that contain the answer to user’s question, as well as performing action automation etc. Detecting bounding boxes, particularly on screen contents, is rather rarely available in existing datasets. It also stretches the model’s capabilities. For this task, we recommend evaluating the bounding box detection quality using F1-Score, where two bounding boxes match if their Intersection over Union (IoU) [Rezatofighi et al., 2019] score is higher than 0.1. In addition, the recognized text representation can be evaluated for each match using Exact match and F1-score.

The rather low threshold is justified because bounding boxes in the dataset are from two sources: VH and manually drawn. VH bounding boxes tend to be big, capturing significant amount of no-content area around the ground truth content. Meanwhile, the manually drawn ones are usually very well fit to the content. When different approaches are used for the same UI element annotation by different raters, their IoU can be very small.

| Question      | SQA-S answer | SQA-L answer            | SQA-UIC answer                  |
|---------------|--------------|-------------------------|---------------------------------|
| What’s the time? | 10:00 a.m.   | The time is 10 a.m.     | [*10:00’, ‘AM’]               |
| What is the date? | 05.06.2024   | The date is June 5, 2024. | [‘05’, ‘06’, ‘2024’ ]         |

See Table 2 for a visualization of the proposed tasks.

6 Baselines

In this section, we present baseline model performance on the tasks introduced in Section 5. The introduced metrics capture several dimensions of model performance and what our dataset enables in terms of downstream applications. We encourage additional research for both improving performance on these tasks, as well as developing additional tasks that leverage the rich screen annotations.

6.1 Experimental Setup: Models

The previously introduced tasks are using the same train, validation and test splits. For each task, we report fine-tuning quality on the described inputs and outputs. Each experiment is ran individually on the two models, which we describe in further detail below.

PaliGemma 3B  The recently introduced VLM leverages the SigLIP loss from PaLI-3 [Chen et al., 2023], while building on top of the Gemma model [Mesnard et al., 2024] as the language backbone. In our experiments we make use of the pre-trained 3B model checkpoint with 896 × 896 resolution. We follow the standard fine-tuning process available publicly. Fine-tuning runs for 10 epochs with a learning rate 1e − 5 using adam optimizer with cosine decay schedule. Both vision and language backbones are trained during fine-tuning.
### 6.2 Experimental Results

We report our learnings in Table 3. We note the slightly higher performance of ScreenAI and attribute it to its larger model capacity and specialized pre-training mixture that includes a richer variety of UI elements. PaliGemma performance is however very competitive, and by specializing both modality backbones we enable better use of the entire model capacity. Task metrics introduced in Section 5 measure performance of models in extracting relevant information for answering a question, ability to provide fluent answers and identify relevant UI elements through their bounding box coordinates. We noticed the first two capabilities correlate with a VLM’s reasoning capacity, and found zero-shot evaluation on open source models of similar size to be much lower in performance (see Appendix E).

Comparing ScreenAI and PaliGemma, we observe that ScreenAI-5B results are 1-2% higher across metrics on SQA-S, SQA-L and SQA-UIC. The difference is even more noticeable for SQA-UIC-BB, with ScreenAI 5.4% higher in BBOX-F1 and 5.2% higher in EM. It appears that while PaliGemma is as good or maybe better at reasoning (counting), understanding UI elements (UI content prediction), it is worse at bounding box localization, and interpreting the question. This could be in part due to its newer Gemma language backbone with more general pre-training, including math datasets. For more qualitative comparisons, please see the Appendix E.

### 7 Conclusion

We introduced ScreenQA, a rich dataset that enables training and evaluating models on question-answering tasks on screen content. We described the annotation process, statistics of the collected dataset, which contains 85,984 question-answer pairs. In addition to answers, our dataset contains extensive annotations of the UI elements, enabling the ability to train or probe models for their holistic understanding of the screen, a necessary capability for high-level reasoning and automation using UI interfaces. Compared to other vision-language tasks, such as document understanding or visual-question answering, the four constructed tasks on the ScreenQA dataset pose unique challenges: rich in text, diverse in mobile applications, blended with icons and symbols. The tasks not only evaluate content quality, but also UI element identification quality. Furthermore, we provided initial results on two model flavors, ScreenAI and PaliGemma, which are best in their parameter class on general document and screen understanding tasks. We further encourage the community to tackle screen content understanding challenges present in our benchmark, to enable new technologies and user experiences.

### References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. PaLM 2 technical
Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărubu, Jason Lin, Jindong Chen, and Abhanshu Sharma. 2024. Screenai: A vision-language model for UI and infographics understanding.

Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, and Blaïs Aguera y Arcas. 2021. UIBert: Learning generic multimodal representations for UI understanding.

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağınak Taşırlar. 2023. Introducing our multimodal models.

Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A. Plummer. 2022. A dataset for interactive vision language navigation with unknown command feasibility. In European Conference on Computer Vision (ECCV).

Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Podlasek, Daniel Salz, Xi Xiong, Daniel Vlasic, Filip Pavetic, Keran Rong, Tianli Yu, Daniel Keysers, Xiaohua Zhai, and Radu Soricut. 2023. Pali-3 vision language models: Smaller, faster, stronger.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST ’17, pages 845–854, New York, NY, USA. Association for Computing Machinery.

Model development contributors, Lucas Beyer*, Andreas Steiner*, André Susano Pinto*, Alexander Kolesnikov*, Xiao Wang*, Xiaohua Zhai*, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, and et al. 2024. PaliGemma.

Gemini Team Google. 2023. Gemini: A family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenheng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. 2023. Cogagent: A visual language model for gui agents. arXiv preprint arXiv:2312.08914.

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

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. arXiv preprint arXiv:2401.13649.

Gang Li, Gilles Baechler, Manuel Tragut, and Yang Li. 2022. Learning to Denoise Raw Mobile UI Layouts for Improving Datasets at Scale. In CHI Conference on Human Factors in Computing Systems, pages 1–13, New Orleans LA USA. ACM.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020a. Mapping Natural Language Instructions to Mobile UI Action Sequences. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8198–8210, Online. Association for Computational Linguistics.

Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. 2020b. Widget Captioning: Generating Natural Language Description for Mobile User Interface Elements. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5495–5510, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries] In Text Summarization Branches Out, pages 14–81, Barcelona, Spain. Association for Computational Linguistics.
Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. arXiv preprint arXiv:2203.10244.

Minesh Mathew, Viraj Bagal, Ruben Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1697–1706.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 2200–2209.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, et al. 2024. Gemma: Open models based on gemini research and technology.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In 2019 international conference on document analysis and recognition (ICDAR), pages 947–952. IEEE.
Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengji Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, and Ying Xiao. 2019. Towards unconstrained end-to-end text spotting. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016a. 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, Austin, Texas. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016b. SQuAD: 100,000+ questions for machine comprehension of text.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8317–8326.

Srinivas K. Sunkara, Maria Wang, Lijuan Liu, Gilles Baechler, Yu-Chung Hsiao, Jindong Chen, Abhanshu Sharma, and James W. Stout. 2022. Towards Better Semantic Understanding of Mobile Interfaces. In Proceedings of the 30th International Conference on Computational Linguistics, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. 2021. Screen2Words: Automatic Mobile UI Summarization with Multimodal Learning. In The 34th Annual ACM Symposium on User Interface Software and Technology, UIST ’21, pages 498–510, New York, NY, USA. Association for Computing Machinery.

Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. 2023. Webui: A dataset for enhancing visual ui understanding with web semantics. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pages 1–14.

Xiaoxue Zang, Ying Xu, and Jindong Chen. 2021. Multimodal icon annotation for mobile applications. In Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction, pages 1–11.
A Data annotation interfaces for question and answer collection

A.1 Question annotation interface

The question annotation interface is shown in Figure 7. Question annotation was performed in a sequential manner by multiple raters. An annotator can see all previous questions to diversify question framing and avoid duplication. We also used the same sequential process to provide more feedback and training to the annotators for quality improvement.

A.2 Answer annotation interface

The answer annotation interface is shown in Figure 8. Answer annotators were tasked to determine if the question is valid and if the question is answerable from the screen context. If both are positive, the annotators need to answer the questions by 1) selecting or drawing the bounding boxes of UI elements, 2) filling the text for each selected/drawn bounding box on right right, 3) ranking them appropriately, 4) providing the full-sentence answer to the question. The annotators were also tasked to review and make necessary corrections if the question has grammatical errors or typos.

B ScreenQA short answers generation prompts

We describe below the prompts used in a version of the PaLM 2 model (Anil et al., 2023) to generate short answers for ScreenQA.

Text information from the ScreenQA dataset (question, list of UI elements descriptions and full-sentence answer) was the input, see B.1 and B.2 for details about the used prompts.
B.1 For answers contained in a single UI element

List various ways to rephrase the answer. The answer should be as short as possible, without extra words from the question. Use all provided elements in each answer. Provide the output in square brackets.

Here is an example:

Question: 'What’s the percentage of humidity?'
Answer elements: ['65%']
Full answer: 'The humidity is 65%'
Rephrases: ['65%']

Here is another example:

Question: 'What is the gender?'
Answer elements: ['Male']
Full answer: 'The gender is male.'
Rephrases: ['male']

Here is another example:

Question: 'What is the status of “24 hr clock”?'
Answer elements: ['on']
Full answer: 'The status is “on”.'
Rephrases: ['on', 'enabled']

Here is another example:

Question: 'What is the age limit for the profile?'
Answer elements: ['18+']
Full answer: 'The age limit of the profile is 18 years or older.'
Rephrases: ['18+', '18 and older', '18 and above', '18 years and above', '18 years old and older']

Here is another example:

Question: 'How many "YPanners" are going to "Daytime Tour At The National Leprechaun Museum"?'
Answer elements: ['35+']
Full answer: 'At least 35 "YPlanners" are going to "Daytime Tour At The National Leprechaun Museum".'
Rephrases: ['35+ ', '35 or more', 'at least 35']

Here is another example:
Question: 'How many items are there in "All Streams"?'
Answer elements: ['1 ']
Full answer: 'There is 1 item in "All Streams".'
Rephrases: ['1 ', 'one ']

Here is another example:
Question: 'What is the status of "Automatic Refresh"?'
Answer elements: ['off ']
Full answer: 'The status of "Automatic Refresh" is "off".'
Rephrases: ['off ', 'disabled ']

Here is another example:
Question: 'What is the location?'
Answer elements: ['Gaylord Opryland Resort Nashville, TN']
Full answer: 'The address is Gaylord Opryland Resort, Nashville, TN.'
Rephrases: ['Gaylord Opryland Resort, Nashville, TN', 'Gaylord Opryland Resort, Nashville, Tennessee ']

Here is another example:
Question: 'What is the application name?'
Answer elements: ['Nails.Makeup.Hairstyle ']
Full answer: 'The name of the application is "Nails.Makeup.Hairstyle".'
Rephrases: ['Nails.Makeup.Hairstyle ']

Here is another example:
Question: 'Where is the store located?'
Answer elements: ['Boston, MA ']
Full answer: 'The store is in Boston, Massachusetts.'
Rephrases: ['Boston, MA', 'Boston, Massachusetts ']

Here is another example:
Question: 'Which tab is selected?'
Answer elements: ['ATP World Tour ']
Full answer: 'The selected tab is "ATP World Tour".'
Rephrases: ['ATP World Tour', '"ATP World Tour" tab ']

Here is another example:
Question: 'When was the "Feeling" post published?'
Answer elements: ['13 hours ago']
Full answer: 'It was published 13 hours ago.'
Rephrases: ['13 hours ago ']

Here is another example:
Question: 'What is the temperature on Friday?'
Answer elements: ['0° ']
Full answer: 'The temperature on Friday is a high of 3° and a low of 0°.'
Rephrases: ['0° ']

Here is another example:
Question: 'What is the maximum and minimum temperature in Western Switzerland on Monday?'
Answer elements: ['−4° ']
Full answer: 'The maximum and minimum temperatures in Western Switzerland are −1° and −4°, respectively.'
Rephrases: ['−4° ']
degrees', 'minimum −4\degree and maximum −1\degree', 'minimum −4 degrees and maximum −1 degrees']

Here is another example:
Question: 'What is the name of application?'
Answer elements: ['babycenter®']
Full answer: 'The application is named "babycenter".'
Rephrases: ['babycenter']

Here is another example:
Question: 'What is the app name?'
Answer elements: ['popcornflix™']
Full answer: 'The name of the application is "popcornflix".'
Rephrases: ['popcornflix']

Here is another example:
Question: 'What is the support email address?'
Answer elements: ['support@stonekick.com']
Full answer: 'The support email address is support@stonekick.com.'
Rephrases: ['support@stonekick.com']

Here is another example:
Question: 'What's the currently playing track name?'
Answer elements: ['Vibe Step']
Full answer: 'The name of the currently playing track is "Vibe Step".'
Rephrases: ['Vibe Step', '"Vibe Step" track']

Here is another example:
Question: 'What is the capital of the Aland Islands?'
Answer elements: ['Mariehamn']
Full answer: 'The capital of the Aland Islands is Mariehamn.'
Rephrases: ['Mariehamn']

Here is another example:
Question: 'What is Central European Standard Time in Albania?'
Answer elements: ['GMT+1:00']
Full answer: 'The Central European Standard Time in Albania is GMT+1:00.'
Rephrases: ['GMT+1:00', '1 hour ahead of GMT']

Here is another example:
Question: 'How many times a week does the activity need to be performed to succeed?'
Answer elements: ['3']
Full answer: 'The activity needs to be performed 3 times a week to succeed.'
Rephrases: ['3', 'three', '3 times per week', '3 times a week', 'three times per week', 'three times a week']

Here is another example:
Question: 'Who is the singer of "Dirty Sprite 3"?'
Answer elements: ['Amero Shotta']
Full answer: 'The singer of "Dirty Sprite 3" is Amero Shotta.'
Rephrases: ['Amero Shotta']

Here is another example:
Question: 'Is there any FastPass available for "Turtle Talk With Crush"?'
Answer elements: ['NO FASTPASS AVAILABLE']
Full answer: '"Turtle Talk With Crush" has no available FastPass.'
Rephrases: ['no', 'no FastPass available']

Here is another example:
Question: 'How many views in total are shown on the video "How to Play Indoor Soccer"?'
Answer elements: ['47,998']
Full answer: 'There are 47,998 shown views in total on the video "How to Play Indoor Soccer".'

Rephrases: ['47,998', '47998']

Here is another example:
Question: 'What can we search for in "Search Homes"?'
Answer elements: ['city', 'zip', 'beds', 'bath', 'price']
Full answer: 'You can search for "city, zip, beds, bath, price" in "Search Homes".'
Rephrases: ['city', 'zip', 'beds', 'bath', 'price']

Here is another example:
Question: 'Which exam was held on April 11, 2016?'
Answer elements: ['IBPS RRB PO PRE: Memory Based Set']
Full answer: 'The exam held on April 11, 2016, is "IBPS RRB PO PRE: Memory Based Set".'
Rephrases: ['IBPS RRB PO PRE: Memory Based Set', '"IBPS RRB PO PRE: Memory Based Set" exam']

Here is another example:
Question: 'What do we need to do to save more?'
Answer elements: ['join VIP']
Full answer: 'To save more, you need to join VIP.'
Rephrases: ['join VIP']

Here is another example:
Question: 'What option is shown in "Middle"?'
Answer elements: ['Pressure']
Full answer: 'The shown "Middle" option is "Pressure".'
Rephrases: ['Pressure', '"Pressure" option']

Here is another example:
Question: 'What day is April 8, 2017?'
Answer elements: ['Saturday']
Full answer: 'April 8, 2017 is Saturday.'
Rephrases: ['Saturday']

Here is another example:
Question: 'What is calculated using the kg unit?'
Answer elements: ['Weight']
Full answer: 'The kg unit is used to calculate weight.'
Rephrases: ['Weight']

Now is your turn.
Question: 'THE QUESTION'
Answer elements: ['THE UI ELEMENT DESCRIPTION']
Full answer: 'THE FULL-SENTENCE ANSWER'
Rephrases:

B.2 For answers contained in multiple UI elements

List various ways to rephrase the answer. The answer should be as short as possible, without extra words from the question. Use all provided elements in each answer. Provide the output in square brackets.

Here is an example:
Question: 'What's the temperature?'
Answer elements: ['59', '"F"']
Full answer: 'The temperature is 59 degrees Fahrenheit.'
Rephrases: ['59°F', '59 Fahrenheit', '59 degrees Fahrenheit']

Here is another example:
Question: 'What is the name?'
Answer elements: ['Jon', 'Brown']
Full answer: 'The name is Jon Brown.'
Rephrases: ['Jon Brown']

Here is another example:
Question: 'What is the rest interval duration?'
Answer elements: ['00', ':', '34']
Full answer: 'The rest interval lasts 00:34.'
Rephrases: ['00:34', '34 seconds', '0 minutes and 34 seconds', '34 minutes', '0 hours and 34 minutes']

Here is another example:
Question: 'What accounts can I use to sign up?'
Answer elements: ['Facebook', 'Twitter']
Full answer: 'You can sign up with "Facebook" and "Twitter".'
Rephrases: ['Facebook, Twitter', 'Facebook and Twitter', 'Facebook or Twitter']

Here is another example:
Question: 'What are the options available for sharing?'
Answer elements: ['Facebook', 'Enjin', 'Email', 'Fake GPS - Search location', 'Android Beam', 'Bluetooth', 'Messaging']
Full answer: 'The available sharing options are "Facebook", "Enjin", "Email", "Fake GPS - Search location", "Android Beam", "Bluetooth", and "Messaging".'
Rephrases: ['Facebook', 'Enjin', 'Email', 'Fake GPS - Search location', 'Android Beam', 'Bluetooth', 'Messaging', 'Facebook', 'Enjin', 'Email', 'Fake GPS - Search location', 'Android Beam', 'Bluetooth' and 'Messaging']

Here is another example:
Question: 'What are the available questions?'
Answer elements: ['Why Michael Flynn kept his Job 17 days after the White House!', 'Caring makes girls run away?']
Full answer: 'The available questions are "Why Michael Flynn kept his Job 17 days after the White House!" and "Caring makes girls run away?".'
Rephrases: ['Why Michael Flynn kept his Job 17 days after the White House!', 'Caring makes girls run away?']

Here is another example:
Question: 'Which tournaments are scheduled from February 20 to 26?'
Answer elements: ['Rio de Janeiro', 'Delray Beach']
Full answer: 'The tournaments scheduled from February 20 to February 26 are "Rio de Janeiro" and "Delray Beach".'
Rephrases: ['Rio de Janeiro', 'Delray Beach', 'Rio de Janeiro and Delray Beach']

Here is another example:
Question: 'How many visitors does "bobbyjones14" have?'
Answer elements: ['1', '3']
Full answer: 'The user "bobbyjones14" has 3 views and 1 view.'
Rephrases: ['1 and 3', 'one and three', '1, 3', 'one, three']

Here is another example:
Question: 'What are the available options in "Most popular"?'
Answer elements: ['United States', 'United Kingdom', 'India', 'Canada', 'Australia', 'Nepal']
Full answer: 'The available options are "United States", "United Kingdom", "India", "Canada", "Australia" and "Nepal".'
Rephrases: ['United States', 'United Kingdom', 'India', 'Canada', 'Australia', 'Nepal']
Here is another example:
Question: 'What is the winning number for April 8, 2017?'
Answer elements: [23, 36, 51, 53, 60, 15]
Full answer: 'The winning numbers for April 8, 2017, are 23 – 36 – 51 – 53 – 60 – 15.'
Rephrases: [23 – 36 – 51 – 53 – 60 – 15', '23, 36, 51, 53, 60, 15']

Here is another example:
Question: 'What are the recent searches?'
Answer elements: ['Hong Kong, Hong Kong', 'SFO ⇄ ORD']
Full answer: 'The recent searches are "Hong Kong, Hong Kong" and "SFO ⇄ ORD".'
Rephrases: ['"Hong Kong, Hong Kong", "SFO ⇄ ORD"', '"Hong Kong, Hong Kong" and "SFO ⇄ ORD"']

Here is another example:
Question: 'Which two countries are playing live?'
Answer elements: [IND, BAN]
Full answer: 'The two countries that are playing live are India and Bangladesh.'
Rephrases: ['India, Bangladesh', 'India and Bangladesh', 'IND, BAN', 'IND and BAN']

Now is your turn.
Question: 'THE QUESTION'
Answer elements: ['THE FIRST UI ELEMENT DESCRIPTION', ...]
Full answer: 'THE FULL-SENTENCE ANSWER'
Rephrases:

C Data examples

Tables 4 and 5 contain a few examples from the ScreenQA dataset. Note that bounding boxes of selected UI elements are highlighted on the screenshot, but they are not actually present in the corresponding image from RICO (Deka et al., 2017).

D Dataset Analysis

![Figure 9: Histogram for the types of questions.](image)

In this section, we show some additional analysis of the collected data.

Table 6 shows the detailed question distribution by categories. Alternatively, Figure 9 shows distribution of question types regardless of the subject.
Table 4: Examples from ScreenQA dataset

Question: ‘When was the match held at Kent State?’
UI elements list:
• [09/24]
Full-sentence answers:
• The match was held at Kent State on September 24.
• The match was held on September 24.
Generated list of short answers:
• 09/24
• September 24
• September 24th
• 9/24

Question: ‘What is the birth date of the user?’
UI elements list:
• [1999], [January], [1]
• [January], [1], [1999]
• [1], [January], [1999]
Full-sentence answers:
• The birth date of the user is January 1, 1999.
• The user’s birth date is January 1, 1999.
• The birth date is January 1, 1999.
Generated list of short answers:
• 1/1/1999
• January 1, 1999
• 1 January 1999
• 1 January, 1999
• January 1st, 1999

Question: ‘What is the status of “Open links inside the app”?’
UI elements list:
• [off]
Full-sentence answers:
• The status of “Open links inside the app” is “off”.
• The status is “off”.
Generated list of short answers:
• off
• disabled
Table 5: Examples from ScreenQA dataset

Question: ‘What is the odometer reading?’
UI elements list:
• [0 m]
Full-sentence answers:
• The odometer reading is 0 m.
• The odometer reading is 0 m.
Generated list of short answers:
• 0 m
• 0 meters

Question: ‘What other applications can be used?’
UI elements list:
• [Android Beam], [Bluetooth]
• [Facebook], [Android Beam], [Bluetooth]
Full-sentence answers:
• The applications that can be used are "Facebook", "Android Beam", and "Bluetooth".
• The other applications that can be used are "Android Beam" and "Bluetooth".
Generated list of short answers:
• Android Beam, Bluetooth
• Android Beam and Bluetooth
• "Android Beam", "Bluetooth"
• "Android Beam" and "Bluetooth"
• Facebook, Android Beam, Bluetooth
• Facebook and Android Beam and Bluetooth
• Facebook or Android Beam or Bluetooth

E Additional Model Evaluation Results

In Section 6 we report the performance of ScreenAI 5B (Baechler et al., 2024) and PaliGemma 3B (development contributors et al., 2024) models on ScreenQA tasks after fine-tuning. It sets the baselines for corresponding model sizes. Here you can find additional evaluations we performed for publicly available models.

E.1 PaliGemma 3B fine-tuning

Pre-trained checkpoints for PaliGemma 3B model are available in 3 resolutions: 224 × 224, 448 × 448 and 896 × 896. To evaluate the influence of input image resolution on model quality, we fine-tuned all 3 checkpoints on all 4 ScreenQA tasks (see Section 5), keeping all other fine-tuning parameters the same (10 epochs, learning rate $1 \times 10^{-5}$, using adam optimizer with cosine decay schedule). You can see the results in the Table 7.
Table 6: Question category distribution and examples.

| Category                  | %   | Examples                                         |
|---------------------------|-----|-------------------------------------------------|
| UI selection & config     | 18.1| Which option is selected? What is the selected ringtone? |
| Quantity number           | 11.7| How many unread messages? How many pictures are there in Western Europe? |
| App name                  | 10.4| What is the name of the application? What is the app name? |
| Date time                 | 9.4 | When was “Heal the Living” released? When is happy hour? |
| Price                     | 3.4 | How much is the gift bonus in 3rd place? What is the price? |
| Name of item              | 3.3 | What is the name of the drug? What is the name of chef? |
| User name                 | 2.8 | What is the name of the user? What is the username on telegram? |
| Duration                  | 2.5 | What is the duration of video? How long is the song? |
| Enum. of avail. options   | 2.5 | Which social media options are given there? What are the options available for logging in? |
| Address and direction     | 2.4 | What is the current location? What is the service zip code? |
| Email address             | 2.4 | What is an email address? What is customer service email? |
| Person’s name             | 2.1 | Who sang the song? What is the last name? |
| Signup/login              | 1.6 | Which application can be used to sign up / login? What are the alternative choices for signing up? |
| Version information       | 1.6 | What is the version number? What is the new feature in version v3.1.3? |
| Weather                   | 1.5 | What is the range of temperature shown on Sunday? What is the weather forecast for Sunday? |
| Score & value             | 1.4 | What is height/weight of the person? What is the score? |
| Yes/No                    | 1.1 | Is there any travel plans? Is there any favorite? |
| Phone number              | 1.0 | What is the phone number? What is the prefix for the international mobile number? |
| # of Stars                | 0.8 | What is the star rating? How many stars are given to the product? |
| Share/sharing             | 0.8 | Which application can be used to share? Where can I share this application? |
| Age                       | 0.8 | How old is …? What is the age? |
| Percentage                | 0.7 | What is the percentage of …? What is the brightness percentage for foreground? |
| Settings                  | 0.6 | What is the setting of …? Which settings are switched on? |
| Quantity amount           | 0.6 | How much fat is there? What is the amount? |
| Permission                | 0.5 | Which application is asking for permissions? What permissions are required for MyCarTracks? |
| # of Likes                | 0.5 | How many likes for …? Which country has the +54 code? |
| Country                   | 0.5 | What is the name of the country? How far is it from …? |
| Distance                  | 0.5 | What is the visibility distance? How many comments? |
| # of Reviews              | 0.4 | What is the number of comments on …? Which is the website address? |
| Website                   | 0.3 | What is the url? What’s the website address? |
| Gender                    | 0.3 | What is the gender? Which gender is displayed on the screen? |
| How to                    | 0.3 | How to start on boot? How to pronounce his name? |
| Currency                  | 0.3 | What is the currency? What is the currency for the price? |
| Unit of measurement       | 0.2 | What is the unit of temperature? What is the unit of weight and length? |
| Language                  | 0.1 | Which language is used in the setting? Which language is being translated into which language? |
| Color                     | 0.0 | What is the UI color? What is the amount of green color? |
| Others                    | 12.8| What’s the average speed? What is the user’s middle initial |
|                           |     | What is the spending limit? Which team has 41 points? |
| Total                     | 100.0|                                                |

Table 7: Performance of fine-tuned PaliGemma 3B models on proposed task types. Bold is best performance.

|                  | SQA-S | SQA-L | SQA-UIC | SQA-UIC-BB |
|------------------|-------|-------|---------|------------|
|                  | EM    | F1    | R-1     | R-2        | R-L        | EM    | F1    | BBOX-F1 | EM    | F1    |
| PaliGemma 3B 896| 89.4  | 93.2  | 90.9    | 85.3       | 90.1       | 86.1  | 87.8  | 88.8    | 78.8  | 81.2  |
| PaliGemma 3B 448| 88.3  | 92.2  | 91.1    | 85.5       | 90.3       | 86.0  | 87.7  | 89.4    | 79.1  | 81.6  |
| PaliGemma 3B 224| 77.5  | 83.9  | 88.2    | 81.5       | 87.4       | 74.8  | 76.7  | 84.9    | 67.5  | 69.6  |

224 × 224 image resolution appears to be too low to capture all the necessary details about the screen. Somewhat unexpected, but there seem to be almost no difference in PaliGemma 3B model performance for resolutions 448 × 448 and 896 × 896.

E.2 Zero-shot evaluations for SQA-S task

Question answering in one form or another is one of the most common tasks for LLMs. This is also true for VLMs. We therefore attempted to evaluate some of those in a zero-shot setting: Fuyu-8b [Bavishi et al., 2023], Gemini 1.5 Flash [Gemini Team Google, 2023], and Gemini 1.5 Pro [Gemini Team Google, 2023], and

https://www.adept.ai/blog/fuyu-8b/
https://deepmind.google/technologies/gemini/flash/
https://deepmind.google/technologies/gemini/pro/
Table 8: Zero-shot performance of public models on SQA-S task. Bold is best performance.

| Model        | SQuAD-EM | SQuAD-F1 |
|--------------|----------|----------|
| Fuyu-8b      | 39.5     | 47.3     |
| Gemini 1.5 Flash | 74.6 | 83.2     |
| Gemini 1.5 Pro  | **79.8** | **86.6** |
| GPT-4o       | 77.8     | 86.6     |

GPT-4o[5](https://openai.com/index/hello-gpt-4o/) (OpenAI et al., 2024). SQA-S task was used for that as the one most similar to other existing benchmarks. Please see Table 8 with all results.

For each model the prompt used in the evaluation is an instruction followed by a question. For the Fuyu-8b model we used the instruction that model’s authors recommended in their examples for similar tasks: “Answer the following DocVQA question based on the image. 

For GPT-4o we tried a couple of different instructions and checked the model output for those on a sample of 10 examples from validation split, and picked the one where results were closest to the expected output: “Answer the question based on the screenshot only. Do not use any other sources of information. The answer should be succinct and as short as possible. If the answer is a text from the image, provide it exactly without rephrasing or augmenting. If there is no answer on the image, output "<no answer>". We don’t claim this is the best prompt to use for this model, nor that we did a very thorough job to find one.

We then re-used this same prompt for the Gemini models evaluation. While the lack of prompt engineering specifically for the Gemini models puts them in disadvantage compared to GPT-4o, results show that even in this setting Gemini 1.5 Pro outperforms GPT-4o by a small margin.

---

[5] https://openai.com/index/hello-gpt-4o/
[6] See https://huggingface.co/adept/fuyu-8b/discussions/28