Baselines and Analysis

Hao Wu* **Jiayi Shen*** **Yanlin Feng*** **Yinghuan Zhang*** **Yuwei Wu***
{haowu3, jiayis2, yanlinf, yinghuan, yuweiwu}@andrew.cmu.edu

1 Models

1.1 Unimodal Baselines

We include the following unimodal baselines: OCR, question text, and detected objects. All these baselines were run using HotpotNet architecture (as detailed in Section 1.2), and only OCR token embedding, or question word embedding, or detected object embedding was passed into the multimodal transformer layer.

Unimodal baseline 1 (OCR) To embed N OCR tokens as $\{x_n^{ocr}\}$, we follow M4C's approach (Hu et al., 2020) and include FastText word embedding, Faster R-CNN appearance features, PHOC character features, and a 4-dimensional location feature:

$$x_n^{ocr} = LN(W_1 x_n^{ft} + W_2 x_n^{fr} + W_3 x_n^{p}) + \text{LN}(W_4 x_n^{b})$$

where W_1, W_2, W_3, and W_4 are learned projection matrices and LN$(\cdot)$ is layer normalization. Different from M4C, the OCR system used is Microsoft Azure OCR. We expect the OCR baseline to capture information provided by scene text that is relevant for answering a given question.

Unimodal baseline 2 (Question text) To embed K question words as $\{x_k^{ques}\}$, we follow M4C's approach and use a pretrained BERT model. We expect the question text baseline to capture information from questions.

Unimodal baseline 3 (Detected objects) We obtain M detected objects through Faster R-CNN, as in M4C. In addition to an appearance feature and a 4-dimensional location feature used by M4C, we incorporate label text, using FastText, into $\{x_m^{obj}\}$:

$$x_m^{obj} = LN(W_5 x_m^{fr} + W_6 x_m^{ft}) + LN(W_7 x_m^{b})$$

where W_5, W_6, and W_7 are learned projection matrices and LN$(\cdot)$ is layer normalization. We expect the detected objects baseline to capture information from image objects that helps answer a given question.

1.2 Simple Multimodal Baselines

Simple multimodal baseline 1 As shown in Fig 1, the first simple multimodal baseline we include is our proposed **HotpotNet**. The architecture is the same as M4C (Hu et al., 2020) but HotpntNet additionally incorporates class label texts into the detected object features. Following Yang et al. (2021), we changed the OCR system from Rosetta-en to Microsoft Azure OCR for better performance. For the detected objects, we thought the background label introduces much noise and thus removed the detected object features with "background" labels. In our actual implementation, we did not use the whole image feature and left it for next steps.

Simple multimodal baseline 2 The second simple multimodal baseline is **HotpotNet without Multimodal Transformer**. In HotpotNet architecture, question text, detected object, OCR token, and previous prediction tokens are encoded with a deep multimodal transformer. With encoded multimodal features, a classifier is used to calculate scores for tokens in fixed vocabulary and a dynamic pointer network is used to dynamically calculate scores for OCR tokens from the image. In this baseline, we explored a simple attention network to model both inter- and intra-modality relations as an alternative for the multimodal fusion layers. Following previous works, we define the *decoding output* as the previous predictions. The encoded decoding output, h^{dec} serves as the input of classifier. Both encoded decoding output, h^{dec}, and encoded OCR representation, h^{ocr}, together serve as the input of DPN. In HotpotNet, a deep unified multimodal transformer

*Everyone Contributed Equally – Alphabetical order

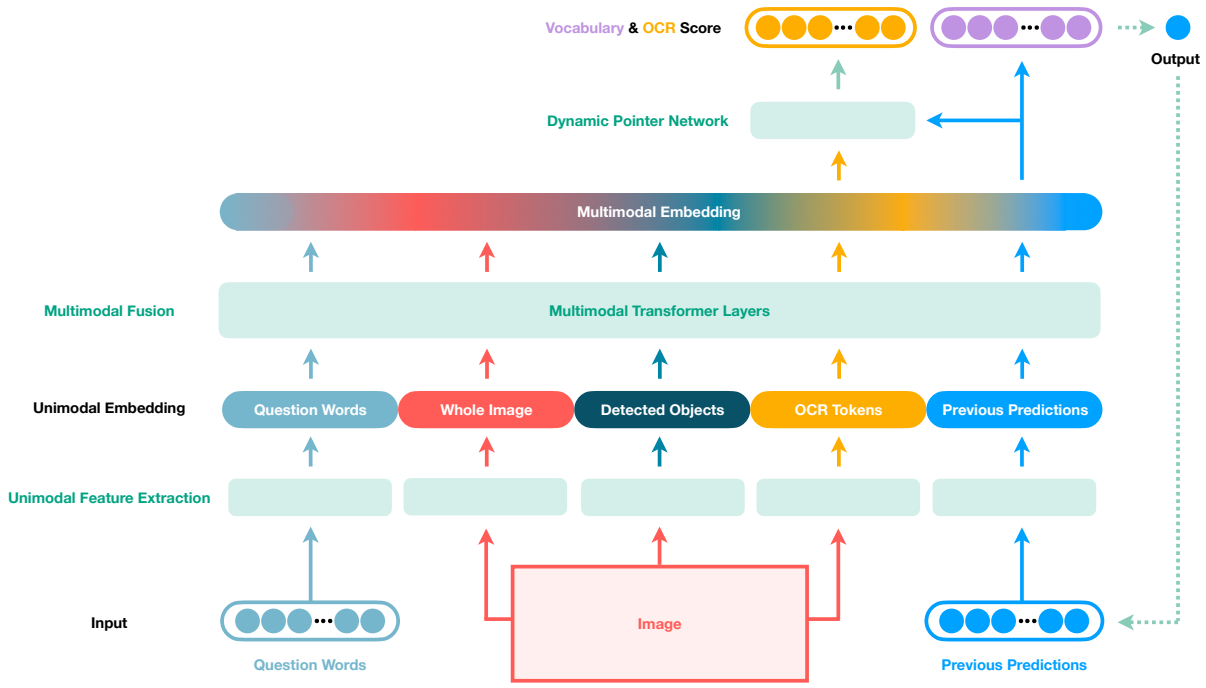


Figure 1: Model architecture of HotpotNet.

encoder is used to get both of the encoded features simultaneously. In this simple baseline, we encode them separately in a pair-wise way.

Three multi-head self attention modules are firstly used to separately model the interaction between previous prediction x^{dec} and each modality embeddings (x^{ocr} for OCR; x^{obj} for detected objects; x^{ques} for question text) :

$$z^{dec\_ques} = \text{SelfAttention}([x^{dec}; x^{ques}])$$
$$z^{dec\_ocr} = \text{SelfAttention}([x^{dec}; x^{ocr}])$$
$$z^{dec\_obj} = \text{SelfAttention}([x^{dec}; x^{obj}])$$

Then the outputs of each self attention module are summed up to get the simple multimodally encoded decoding representation:

$$h^{dec} = z^{dec\_ques} + z^{dec\_ocr} + z^{dec\_obj}$$

The multimodal encoding of OCR token representations is similar:

$$z^{ocr\_ques} = \text{SelfAttention}([x^{ocr}; x^{ques}])$$
$$z^{ocr\_obj} = \text{SelfAttention}([x^{ocr}; x^{obj}])$$
$$h^{ocr} = z^{ocr\_ques} + z^{ocr\_obj}$$

During decoding, for each time step t, the fixed score for the vocabulary is calculated by a classifier:

$$y^{\text{vocab}}_{t,m} = \left(w^{\text{vocab}}_{m}\right)^{T} h^{\text{dec}}_{t} + b^{\text{vocab}}_{m}$$

where w^{vocab}_{m} is a d-dimensional parameter for the m-th wordx.

The copy score for the dynamic pointer network is calculated through the decoding outputs h^{dec}_{t} and OCR token representations h^{ocr}_{n}:

$$y^{\text{ocr}}_{t,n} = (W^{\text{ocr}} h^{\text{ocr}}_{n} + b^{\text{ocr}})^{T} \left(W^{\text{dec}} h^{\text{dec}}_{t} + b^{\text{dec}}\right)$$

where W^{ocr} and W^{dec} are $d \times d$ matrices, and b^{ocr} and b^{dec} are d-dimensional vectors.

Simple multimodal baseline 3 For the third simple multimodal baseline **HotpotNet without Dynamic Pointer Network**, we replaced the dynamic pointer network (DPN) decoder of HotpotNet with the classifier used in LoRRA (Singh et al., 2019). In our implementation, we passed the decoding outputs into the LoRRA's classifier to either select a frequent answer from the training set or copy a single OCR token in the image as the answer.

1.3 Competitive Baselines

We reproduced the performance of three competitive baselines that were previously proposed in

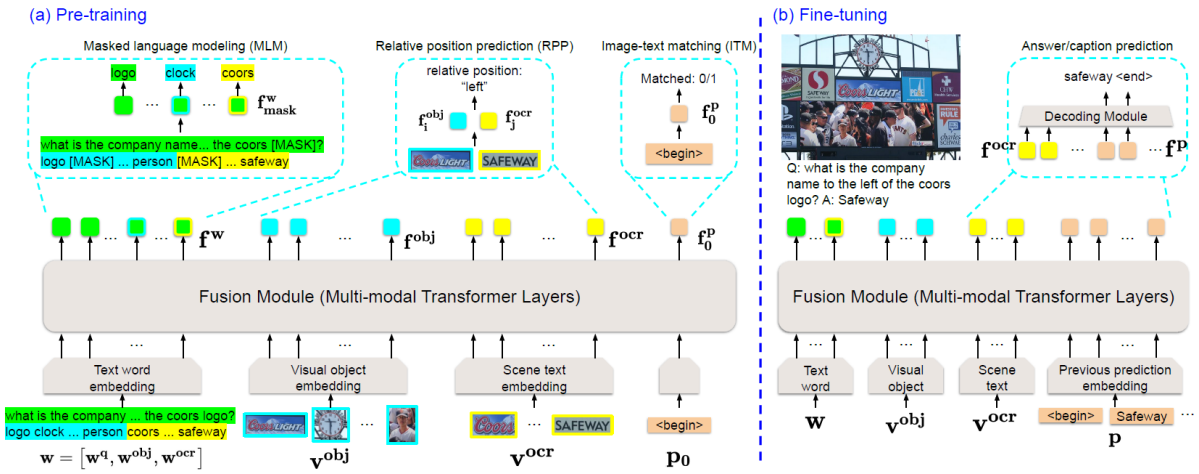


Figure 2: Model architecture of TAP.

research papers, namely LoRRA, M4C and TAP. In this section, we will briefly introduce their model architecture and key insights.

LoRRA Singh et al. (2019) proposed LoRRA that is specifically designed for reasoning about text in images. They proposed a reading module that uses an OCR model to extract word tokens from the image, embeds them with FastText vectors and computes contextual attention based on the question to get the combined representation.

$$f_{OCR}(s,q) = f_{comb}(f_A(f_O(s), f_Q(q)), f_Q(q))$$

LoRRA predicts answer by selecting from a vocabulary of frequent answers and the OCR tokens in the image. Experiments show that the proposed reading module, when used with the previous state-of-the-art VQA model, improves performance significantly on TextVQA and slightly on VQA 2.0.

M4C The major limitation of LoRRA is that it only models interaction between pairs of modalities with pairwise fusion mechanism. M4C (Hu et al., 2020) addressed this with a multimodal transformer architecture that enables fusion of more than two modalities in a common semantic space. They also replaced the answer prediction modules with a dynamic pointer network to enable prediction of multi-token answers.

TAP Yang et al. (2021) further explored unsupervised text-aware pre-training (TAP) with masked language modeling, image-text matching and relative position prediction, and experimental results show that pre-training on TextVQA itself is able to boost performance by $+5.4\%$. The architecture of TAP (Fig 2) is similar to M4C except that the former further includes object labels.

2 Results

2.1 Evaluation Metric

Following previous work (Hu et al., 2020; Yang et al., 2021), we evaluated the models using an accuracy metric based on soft voting. TextVQA provides 10 human-annotated answers for each questions. The accuracy score of a certain answer is computed by averaging the following metric over all 9 out of 10 subsets of annotators.

$$\text{Acc}(ans) = \min\left\{\frac{\#\text{humans that said } ans}{3}, 1\right\}$$

The evaluator also performs text normalizations such as lowercasing and converting number words to digits before evaluting the answers [1].

2.2 Comparison of Unimodal Baselines

Out of the three unimodal baselines, OCR has the highest validation accuracy, and detected objects has the lowest validation accuracy. This aligns with our intuition and the experiment results of LoRRA (Singh et al., 2019). According to Biten et al. (2019), 61% of the answers in TextVQA contain OCR tokens. Therefore, OCR alone provides significant information for the task. Question text also contributes significantly to the task. Question starting words indicate question types (e.g. what, who, is, how...), and question words in general can have correlations with answer words (e.g. "sneakers" and brand names like "Adidas" and "Nike").

[1] For full details of the evaluation, please refer to https://visualqa.org/evaluation

Methods	Dev Accuracy ↑
Unimodal Baselines	
1. OCR	25.26
2. Question text	19.00
3. Detected objects	12.72
Simple Multimodal Baselines	
1. HotpotNet	45.32
2. HotpotNet w/o multimodal transformer	41.38
3. HotpotNet w/o DPN	28.88
Previous Approaches	
1. LoRRA (Singh et al., 2019)	27.57
2. M4C (Hu et al., 2020)	39.23
3. TAP (Yang et al., 2021)	49.26

Table 1: Experiment results of unimodal, multimodal, competitive baselines on TextVQA dev set (Singh et al., 2019).

In contrast, detected objects have the least power in predicting answers, as they do not contain answers themselves and lack knowledge of what the question is asking.

2.3 Comparison of Simple Multimodal Baselines

Out of the three simple multimodal baselines, HotpotNet achieved the best performance on the dev set. Compared with Simple Attention, which uses five self-attention modules to model pair-wise interactions, the multimodal transformer layer treats all feature inputs as one whole sequence and deploys attention over the whole sequence at once. Thus it can capture more and deeper inter- and intra- interactions for all modalities. When the decoder is replaced by the LoRRA's classifier, the performance drops significantly which shows the effectiveness of the dynamic pointer network from M4C.

2.4 Comparison of Competitive Baselines

Out of the three competitive baselines, TAP achieved the best accuracy score on the dev set. This shows the effectiveness of the pre-training strategies proposed by Yang et al. (2021). Compared with M4C and LoRRA, our proposed HotpotNet has significantly better performance. This improvement mainly comes from the OCR system we used, according to our ablation study. We leave more comprehensive analysis in Section 3. Inspired by TAP's results, we will also explore the pre-training strategies for our proposed architecture in next steps.

3 Analysis

3.1 Ablation Study

Apart from testing simple unimodal baselines and tweaking model structures, we have taken a step further to explore the significance of various features by conducting an ablation study, with quantitative results summarized in Table 2.

According to Singh et al. (2019), the accuracy of OCR module is a key determinant of the model performance upper bound on TextVQA tasks. Therefore, based on the M4C competitive baseline, we first substituted Rosetta with the state-of-the-art Microsoft Azure OCR module to feed in more accurate OCR detection data, which corresponds to the last entry of HotpotNet in Table 2. This substitution yielded an increase in validation accuracy from 39.23 (M4C) to 45.37 (Hotpot w/o Object Label or Object Filtering).

Moreover, as proposed in HotpotNet, we are interested in incorporating various information about detected objects including label text, position and visual features.

Accordingly, we first visualized the detected objects on question images, and realized the existence of numerous "noisy" and repetitive detections. As shown in Fig 3, the excessive number of detection boxes labeled as "background" are heavily overlapped with one another and seem irrelevant to the

Model	OCR	Object Filtering	Object Labels	Dev Accuracy(%) ↑
HotpotNet	Microsoft-OCR	✓	✓	45.32
	Microsoft-OCR	✓	✗	45.41
	Microsoft-OCR	✗	✓	44.92
	Microsoft-OCR	✗	✗	45.37
M4C	Rosetta-OCR	✗	✗	39.23

Table 2: Ablation study of feature selection on HotpotNet

image of a can. We were thus motivated to "filter" the object input specifically with the label "background", which resulted in an outcome (45.41) almost the same as what HotpotNet without filtering achieved (45.37). Consequently, the assumption treating "background" objects as noises seem implausible, and further experiments are needed to examine the importance of objects.

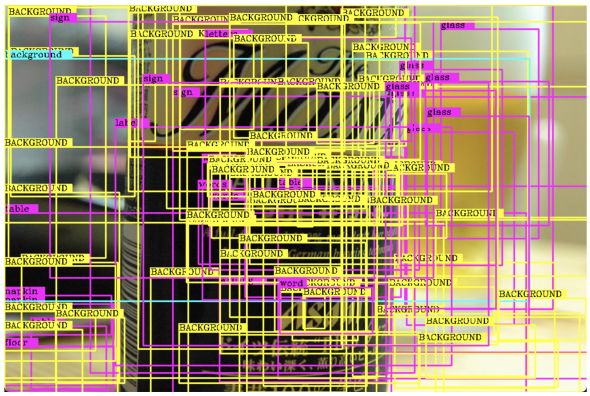


Figure 3: Visualization of object detection.

On the other hand, our hypothesis on the usefulness of object labels seems untenable. Without object filtering, integrating object labels as input brings about an accuracy decline of 0.45 from 45.37 (HotpotNet w/o Object Filtering or Object Labels) to 44.92 (HotpotNet w/o Object Filtering). Sections 3.2 and 3.3 provide further quantitative and qualitative analysis on the effect of object label input.

3.2 Intrinsic Metrics

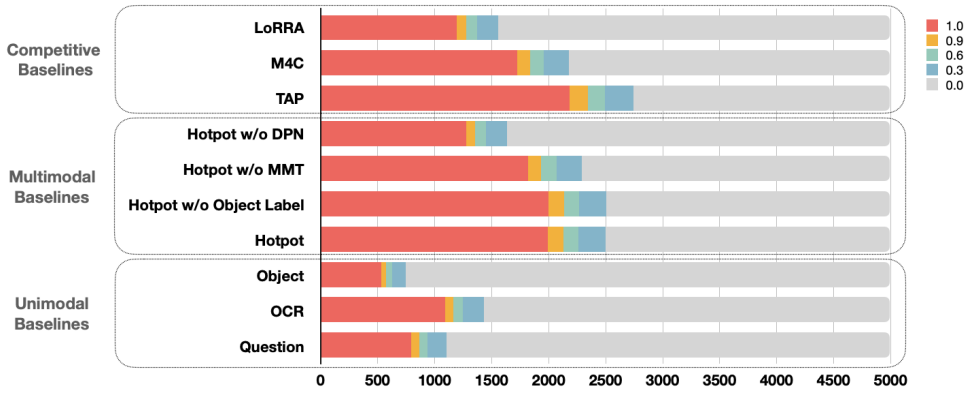
In Fig 4(a), we visualize the overall distribution of accuracy scores on 5000 questions for each model, from which selected qualitative examples will be presented in Section 3.3. As presented in the bar plot, while the state-of-the-art TAP model tops all baselines in all categories, i.e. has the smallest number of 0-scored questions and the largest number of none-zero scored questions, our proposed HotpotNet (and the variant without object label input) also achieved leading performances, despite slightly fewer counts in predictions with 1.0 accuracy. Meanwhile, unimodal baselines as well as multimodal baselines such as LoRRA and Hotpot w/o DPN have unarguably weaker performance. We extracted subsets of interest from validation data to further dissect model performance in the following subsections.

3.2.1 Model Performance on Position Related Questions

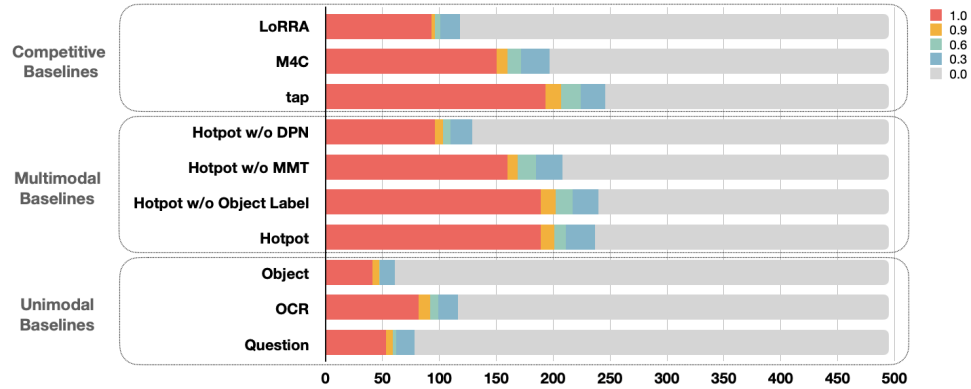
As mentioned in SA-M4C (Kant et al., 2020), $\sim$ 13% of questions in the TextVQA dataset are related to spatial positions. Therefore, the capability to reason about spatial position information is essential in tackling TextVQA tasks. We developed a position-related question subset where the question explicitly contains at least one word from the predefined positional vocabulary list, `<left, right, under, upper, below, above, head, tail>`. We used about 10% of the validation set to construct a 495-sample subset.

We visualized the distribution of accuracy scores in Fig 4(b). In this subset, the performance gap between HotpotNet and the state-of-the-art model, TAP, is much smaller. All models examined perform worse on the position related questions, compared to their performance on all questions. From this perspective, we conclude that existing models have substantial room for improvement in position related questions. For next steps, we plan to use pre-training tasks and data augmentation strategies to enhance the model's ability to understand and answer position related questions.

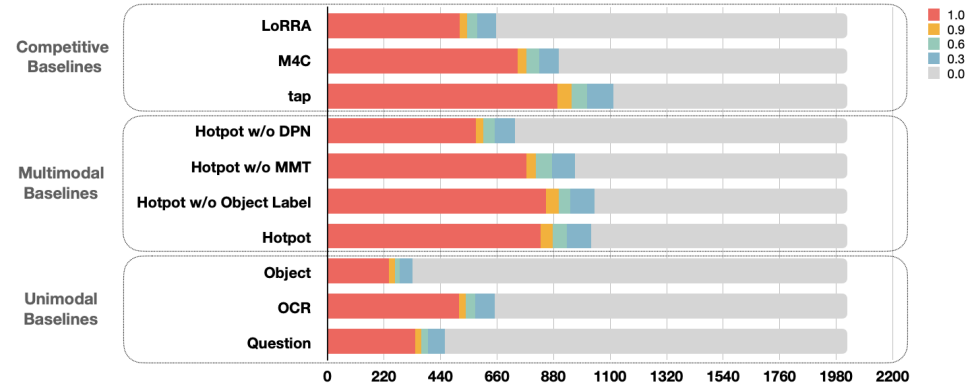
In addition to this subset, we observe questions with position-related semantics, e.g., "pointing to", "the back". We hypothesize that all position related questions, including spatially and semantically, are some of the harder questions for the TextVQA task and deserve attention in future work.



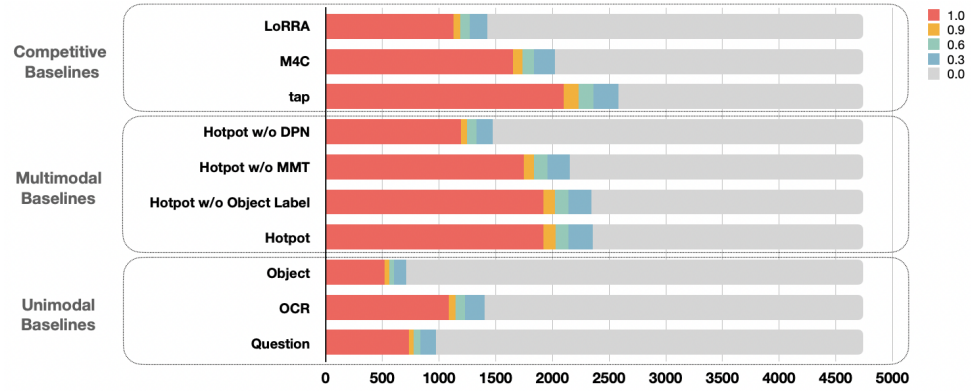
(a) Whole validation set



(b) Questions related to position

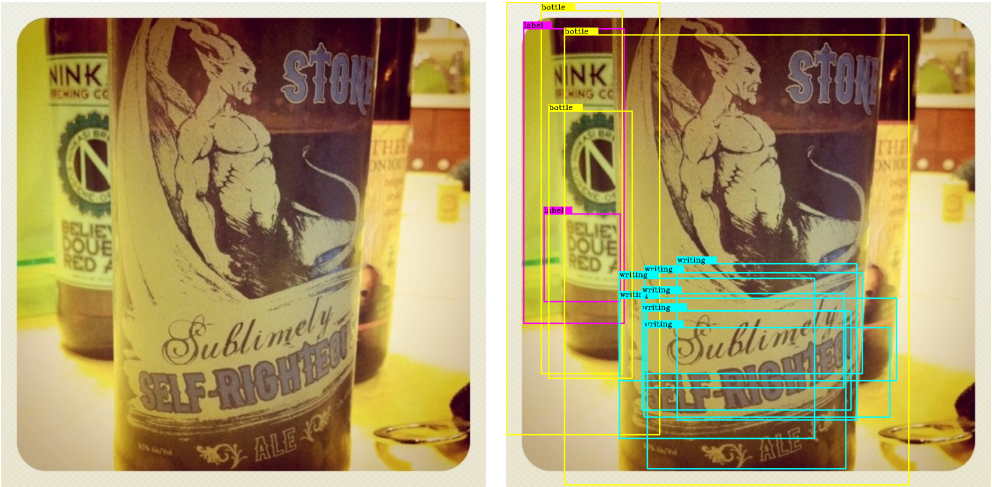


(c) Questions related to object labels



(d) Questions requiring reading

Figure 4: Accuracy distribution in subsets



(a) Example image without bounding boxes

(b) Example image with bounding boxes

Figure 5: Example image with/without bounding boxes

3.2.2 Model Performance on Object Related Questions

TAP (Yang et al., 2021) first incorporates class label text of detected objects into the model by combining it with OCR text and question text to create text word embedding. However, with this approach, the class label text is not paired with or aligned to the corresponding detected object. Our initial hypothesis is that the introduction of class label text can help the model align the visual features of detected objects with their textual counterpart in questions. However, the model's performance worsened after the label text was introduced. Therefore, we created another subset where questions explicitly contain the class label text of at least one detected object in the corresponding image. We used this subset to further explore the effect of introducing object class label text.

The subset of object related questions consists of 2024 images, about 40% of the validation set. This percentage shows the importance of enabling the model to better understand object information. The accuracy score distribution of this subset is shown in Fig 4(c). Surprisingly, the performance of HotpotNet without object label text is better than that of HotpotNet with object label text. In the latter model, we align class label text with its corresponding detected object; when an image has multiple detected objects of the same class, the same class label text is included in each of these objects' features and effectively multiple times in the entire detected object embedding of the sample. For such samples, TAP only includes a class label once in the entire text word embedding. Visualizing the images with detected object bounding boxes, we found several redundant bounding boxes for one object in a given image, as illustrated in Fig 5. Ideally, we would implement bounding box fusion before feeding the bounding boxes into the multimodal transformers. Without this fusion step, 7 different bounding boxes with the same "writing" label text as shown in Fig 5(b) were fed into the model. We conjecture that the noisy pairs of class label text and bounding box confused the model and that including label text of the same class only once promoted an implicit bounding box fusion in TAP.

3.2.3 Model Performance on Questions Requiring Reading Text

TextVQA is a task that requires reading and understanding text in images. However, we found that 5% questions of the validation set does not require reading texts. Therefore, we constructed a subset of questions that require reading texts to answer, consisting of 4743 questions.

Focusing on this subset, even with the noise brought by the multiple object class label texts as mentioned in Section 3.2.2, HotpotNet performs better than HotpotNet without object labels. Specifically, HotpotNet has a smaller percentage of 0.0 accuracy than that of HotpotNet without object labels and larger percentages of 0.3, 0.6, 0.9, and 1.0 accuracies than those of HotpotNet without object labels. We also calculated the average score on the not-require-reading subset, consisting of 257

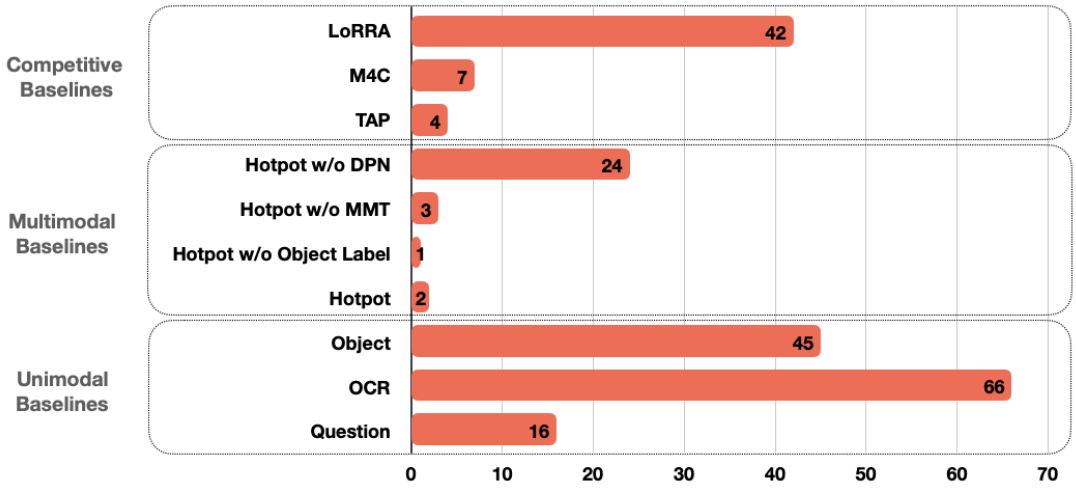


Figure 6: Counts of worst performance. For each model, we count the number of questions for which it got a 0 accuracy score while the others got nonzero scores.

images, where HotpotNet has an average score of 23.28, lower than 25.14, the score of HotpotNet without object labels. This result suggests that the potential of the label text should be further investigated. From this discovery, we hope to figure out a way to leverage the complementary information from label text without introducing the noise.

3.3 Qualitative Analysis and Examples

In this section, we first qualitatively examine model performance by providing one to two representative failure cases of each model and interpreting the observations. Then we summarize findings on subsets of questions that we consider especially challenging.

3.3.1 Case Analysis of All Baselines

For failure examples of all baseline models, we focus on a subset of 210 questions where only one baseline model got 0 accuracy score while the others got nonzero scores, i.e. one model performs worst. We also discuss cases where there are both zero-scored model(s) and perfectly-scored model(s). Both scenarios reflect that a model is weaker than its counterparts.

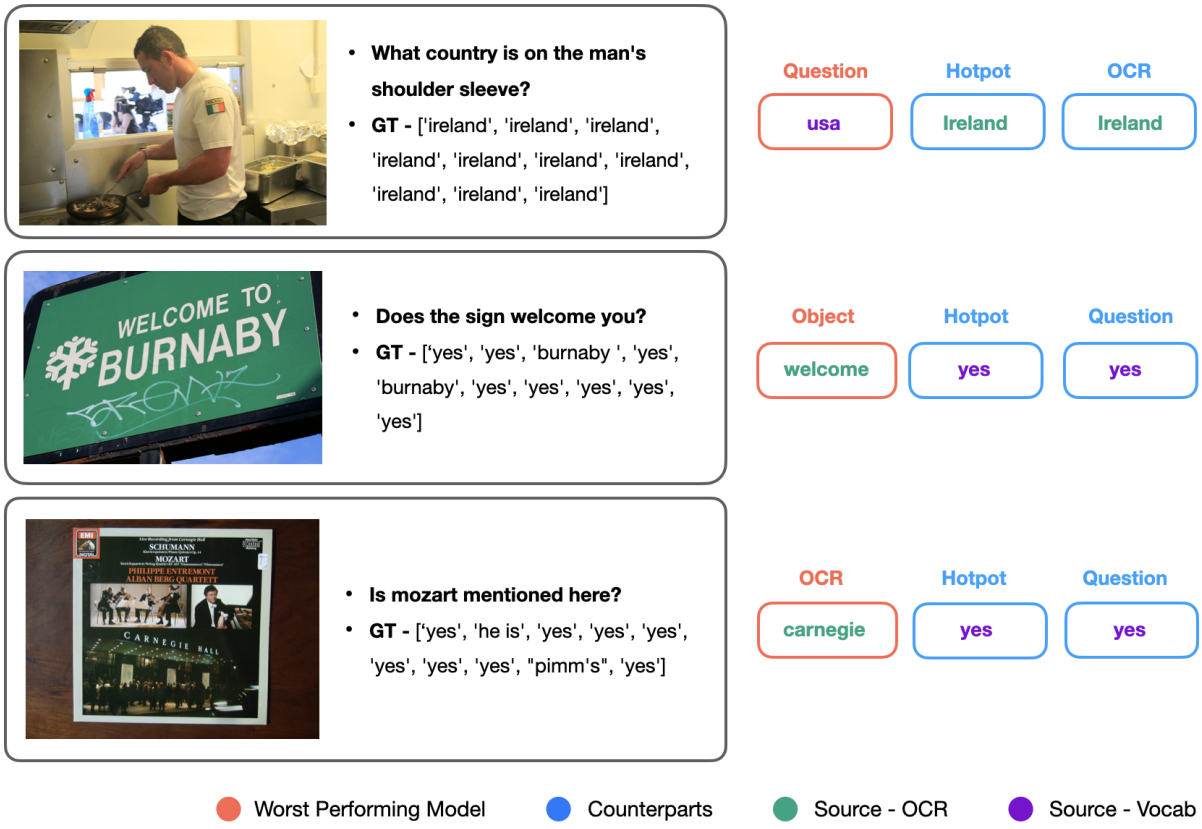
Fig 6 counts the number of questions on which each model performs the worst. According to the plot, multimodal baselines except Hotpot without DPN have the smallest numbers of worst performing questions, and these numbers are comparable with those of the stronger competitive baselines, M4C and TAP. All three unimodal baselines as well as LoRRA have numerous worst performing questions, confirming their overall poor performance on the entire validation set.

Fig 7(a) lists three typical failure cases pertaining to each of the unimodal baselines. In the example of OCR unimodal baseline, as the input only involves OCR tokens, the model has no clue about the question prompt and therefore outputs an OCR token based on its best guess. Similarly, the unimodal baseline of detected object also suffers when the answer cannot be directly derived from the image without the guidance of question text, especially in the case of yes-no questions. In comparison, in both questions, the unimodal baseline of question text and multimodal methods such as HotPotNet and TAP take advantages of information about question and output correct answers. For the unimodal baseline of question text, due to the lack of visual information and OCR token, the model acts as if it was solving a QA question and inferred the answer from the keyword “country” in the question text. On the contrary, multimodal approaches and the OCR unimodal baseline made perfect predictions by taking in OCR information.

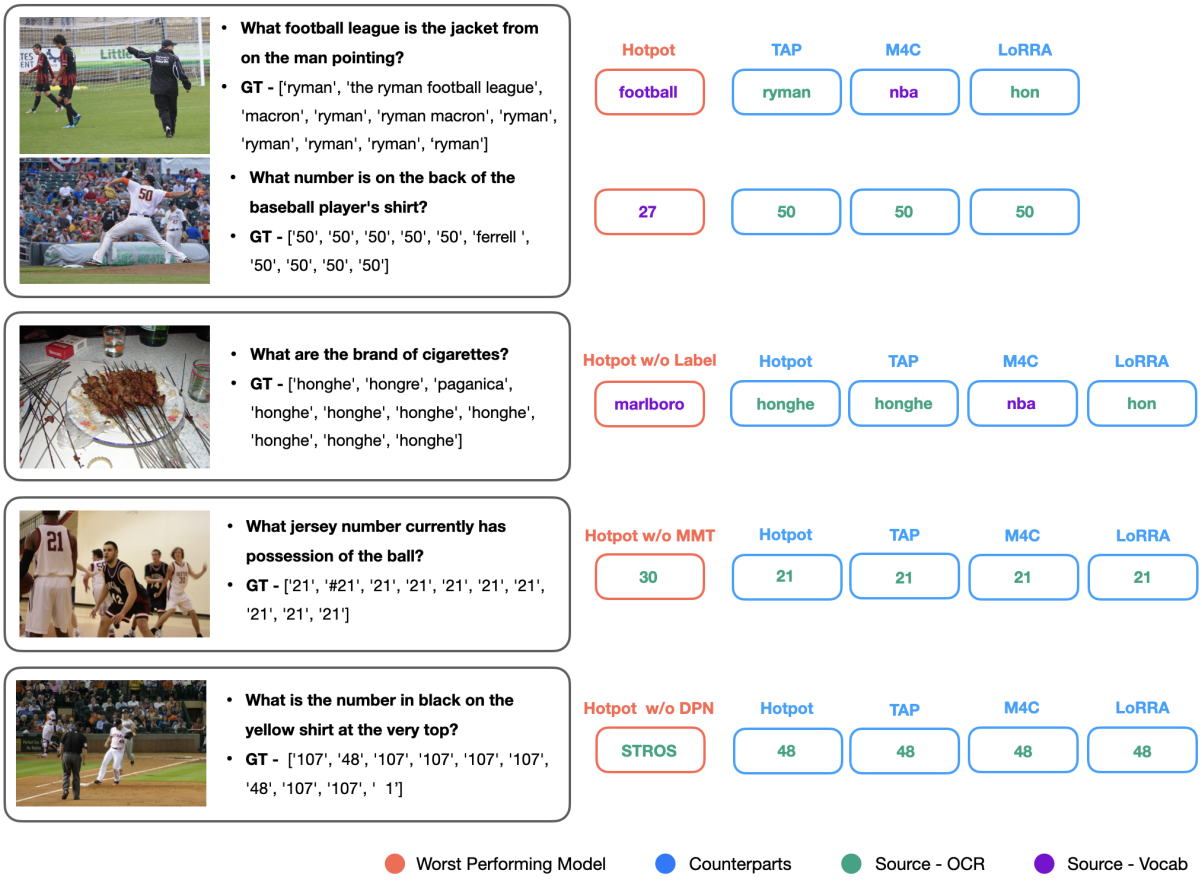
In all three examples above, unimodal baselines failed due to the lack of information from multiple modalities such as text and vision, demonstrating the multimodal nature of the TextVQA task.

Fig 7(b) reveals failure examples of HotpotNet and its variants with tweaks in feature input or model structure. The answers of HotpotNet and its variants are listed in juxtaposition with those from competitive baselines.

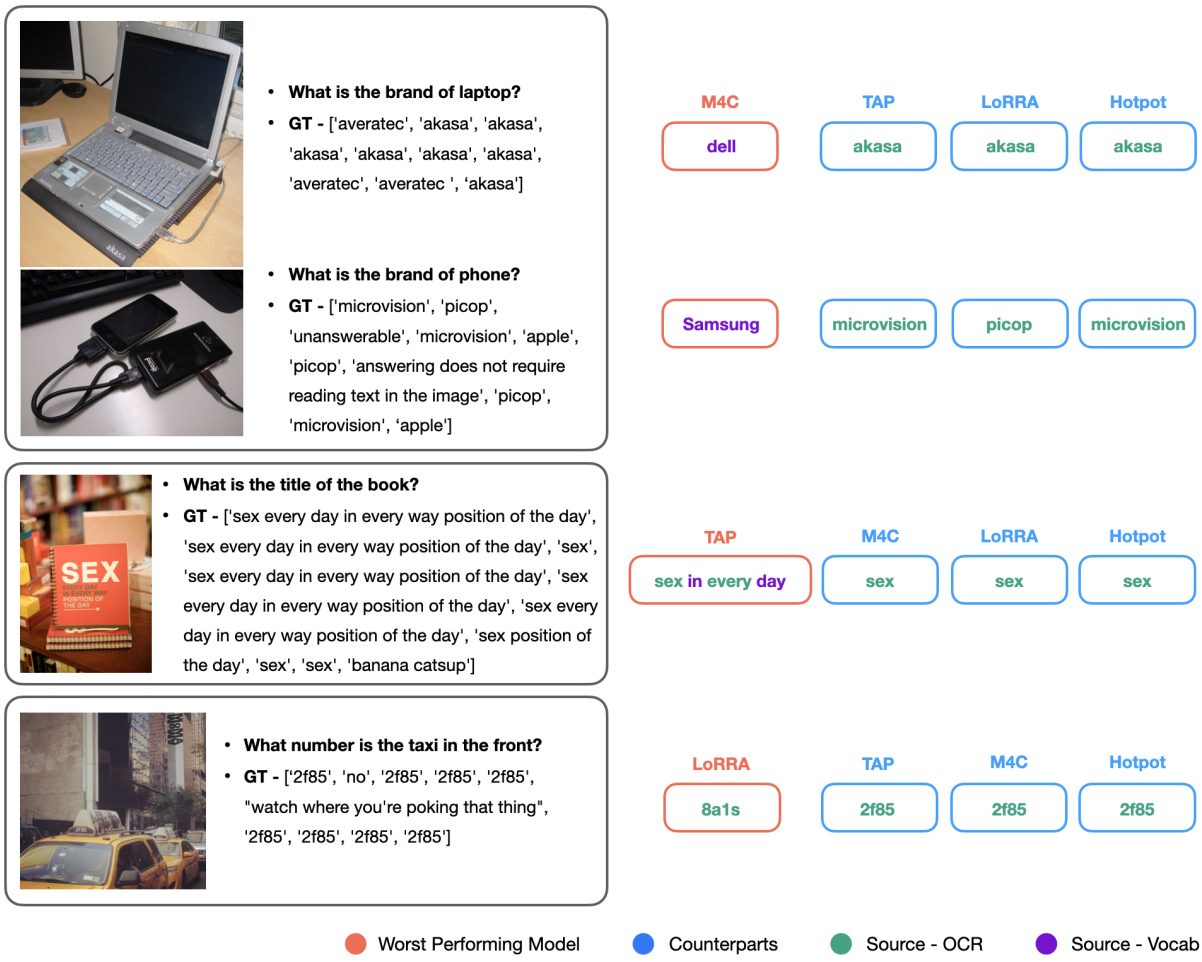
For vanilla HotpotNet, the incorporation of positional information in forms of relative coordinates seems impotent to questions involving complicated



(a) Failure examples of unimodal baselines. We denote unimodal baselines of OCR, question text, and detected objects as "OCR", "Question" and "Object", respectively. We abbreviate the HotpotNet model as "Hotpot".



(b) Failure examples of HotpotNet and variants.



(c) Failure examples of competitive baselines.

Figure 7: Failure examples

positional relationships. In the question of football league, the model is required to figure out the implicit positional information inferred from the keyword "pointing", while the other question asks about a semantically positional relationship denoted by "the back". In both questions, TAP attained correct predictions. This superior performance might be attributed to its pre-training efforts in relative position prediction. Accordingly, our next step is to implement pre-training of position prediction, as included in our 5 proposed pre-training tasks in the previous report.

For HotpotNet with structural changes, in the case of HotpotNet with MLP as decoder, the fact that HotpotNet (equipped with dynamic pointer network as decoder) outputs "48" means token "48" was indeed detected by the MSOCR module and served as an input to all HotpotNet models. The answer "STROS" decoded by MLP indicates that MLP failed in interpreting the multimodal features, especially the keywords in question text ("number", "yellow shirt", etc) and the corresponding visual content. Moreover, in the question asking jersey number, since HotpotNet predicted "21" as answer, "21" was undoubtedly an OCR input to Hotpot w/o DPN. However, without multimodal transformer layers, Hotpot w/o DPN seems weak in attending to the specific region of images for answer, and therefore outputs a word that is close to the ground truth but does not make sense with respect to the question text.

Moreover, from the perspective of feature intake, HotpotNet without object label as input exhibits weak performance on those questions where label text occurs in the question text. In the example of cigarette, Hotpot w/o Label has no way to capture the echoing between label text and question text, which provides hint on which detected object to focus on. Therefore, the model has failed to locate and extract the answer from the image, and resorted

to vocab to find a relevant answer. In comparison, with integration of label information, HotpotNet has been able to find the answer based on OCR scan.

Fig 7(c) presents several cases where competitive baselines have performed the worst.

In both questions asking about the brands of electronic products, M4C has output answers from vocabulary that are semantically close to what the question is asking for. Our interpretation is that M4C is able to reason about the visual information and understand the "phone" and "laptop", but fails to align the OCR tokens to visual part. These cases suggest we work on a deeper and more comprehensive fusion cross modalities.

As for TAP, in all 4 questions where it performs the worst, its predictions are not necessarily wrong from our perspective. In the typical case of book title question, all the other baseline models have output single word "sex", which matches the ground truth in some degree. In comparison, TAP seems to have taken a step further to provide a longer and more complete answer by linking two important OCR tokens ("sex" and "every") with "in" and "day" from vocabulary, which is still semantically meaningful. The observation of TAP has shed light on a more meaningful metrics instead of clear cut-off, which we will leave as next step.

An exemplary failure condition of LoRRA is questions requiring positional reasoning. In the question of taxi number, models are required to focus on the taxi "in the front", which is almost unattainable for LoRRA without incorporating location feature. In comparison, models taking advantage of positional feature input or positional pre-training tasks, such as TAP and HotpotNet, have perfect predictions.

3.3.2 Challenging Cases

In addition to questions where individual model has performed poorly, we are interested in the case where all of our baselines failed, i.e, we are interested in "hard" questions. In fact, 24.8% of the validation dataset turn out to be questions where none of the models outputs predictions matching any ground truth answer. In the following paragraphs, we categorize challenging questions into three groups, and provide interpretation with reference to qualitative examples.

First, reasoning of *positional relationship*, including both spatially positional relationship and semantically positional relationship, is typically challenging for our entire baselines. Following (Kant et al., 2020), spatially positional relationship consists of 12 relative position relationships, including `on`, `cover`, `ovelap`, `eight-way relative orientation`, and `unrelated`. Semantically positional relationship refers to a broad range of relative positions bearing semantic meanings, with examples including `the back`, `pointing to`, `the head`, `the tail`, `the closest` etc. In the question in Fig 8, models are required to figure out the "closest up beverage", which involves reasoning of spatial distance and therefore are particularly difficult.

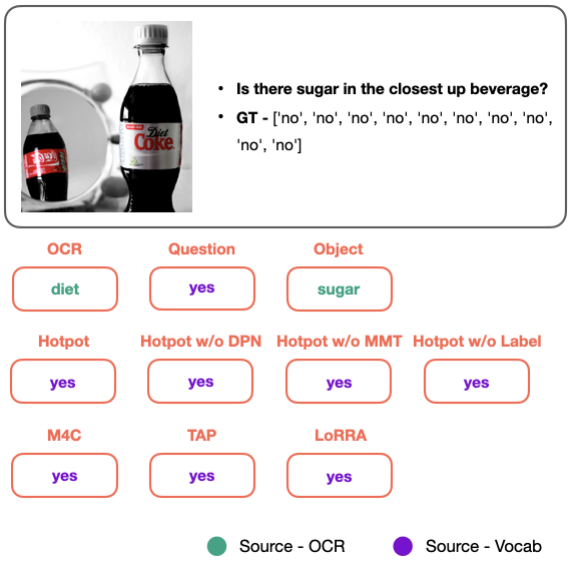


Figure 8: Challenging cases of positional relationship

A second group of challenging cases are those calling for multi-stage reasoning. As in Fig 9, question of 8 clocks asks models whether all the clocks are set for the same time. To derive the correct answer, models are supposed to be able to firstly figure out the time of each clock and then compare the results over all the clocks. Similarly, the question of brewery requires models to not merely detect the brewery of each beer, but also compare the breweries among all beer to solve the puzzle.

Furthermore, there are questions where prior knowledge is essential to the answers but cannot be learned from question text or image. Although a model might be able to learn certain type of knowledge during training, out-of-domain knowledge present in questions would still render the model impotent. In the two examples in Fig 10, neither "roman numerals" nor "lottery" is present in the images, and therefore none of our baselines would

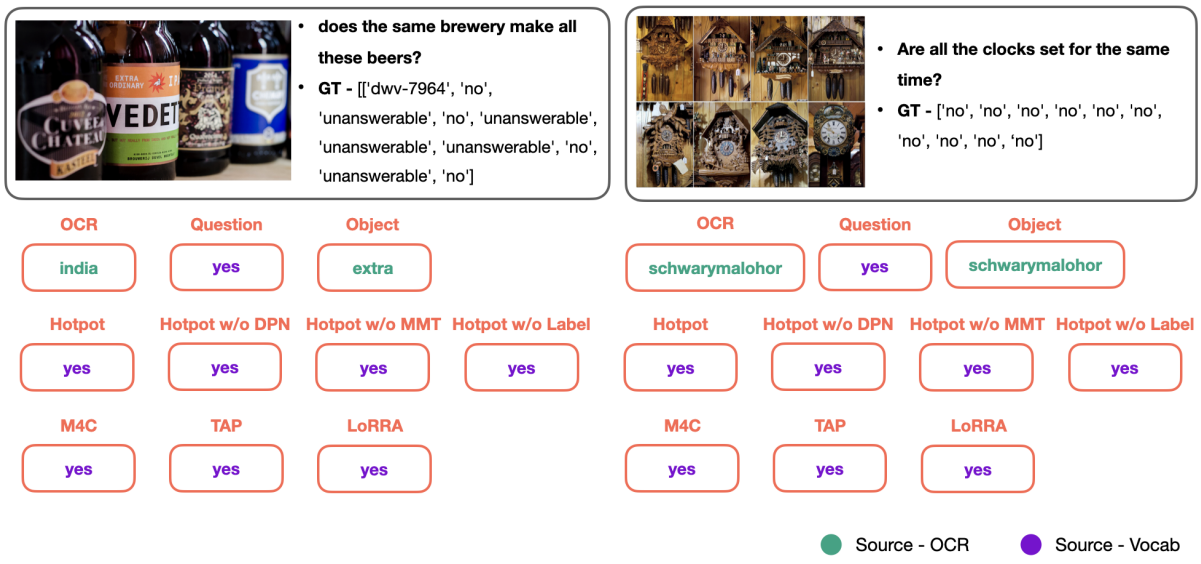


Figure 9: Challenging cases requiring multi-stage reasoning

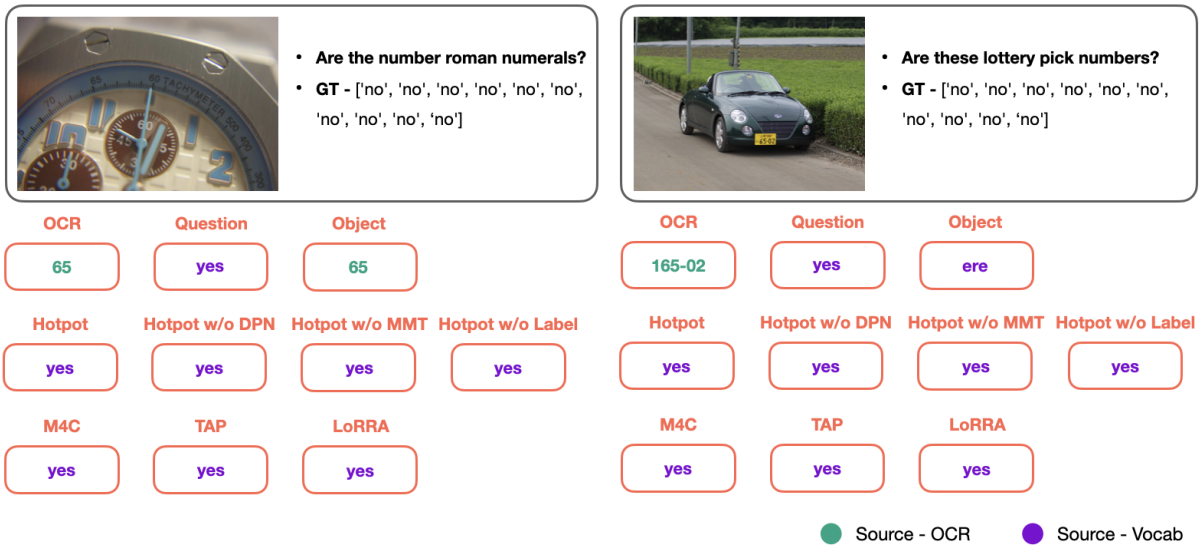


Figure 10: Challenging cases requiring prior knowledge

be capable to predict correct answers as guided by question text "roman numerals" and "lottery".

4 Team member contributions

Yuwei Wu contributed to coding, running experiments, and writing sections on models and results, including unimodal and multimodal baselines.

Jiayi Shen contributed to coding and writing sections on models and results, including unimodal and multimodal baselines.

Yanlin Feng contributed to running competitive baselines and writing sections related to competitive baselines and evaluation metrics.

Yinghuan Zhang contributed to splitting subsets and processing subsets statistics, visualizing statistical and qualitative results, and writing sections on model analysis, intrinsic metrics, and challenging cases.

Hao Wu contributed to processing model output into summary sheets, visualizing statistical and qualitative results, and writing sections on ablation study, intrinsic metrics, and qualitative analysis and examples.

References

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. 2019. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301.

Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. 2020. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9992–10002.

Yash Kant, Dhruv Batra, Peter Anderson, Alexander Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. 2020. Spatially aware multimodal transformers for textvqa. In *European Conference on Computer Vision*, pages 715–732. Springer.

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 (CVPR)*.

Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. 2021. Tap: Text-aware pre-training for text-vqa and text-caption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8751–8761.