

000 001 002 003 004 005 006 007 GQA-Q2Q: A LARGE-SCALE DATASET FOR RE- 008 SOLVING ENTITY AMBIGUITY IN VISUAL QUESTION- 009 ANSWERING VIA CLARIFYING SUB-QUESTION 010 011 012

013 **Anonymous authors**
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053

Paper under double-blind review

ABSTRACT

Vision-Language Models (VLMs) have achieved remarkable results on various visual question-answering (VQA) benchmarks. However, their performance is significantly impacted by ambiguous questions in which the target entity in the image is not clearly identified. To address and evaluate this issue, it is essential to create a dedicated benchmark dataset that aligns ambiguous questions with a clarifying sub-question. However, constructing a large, high-quality dataset is costly, particularly when it relies on expert annotations. To efficiently construct such a dataset at scale, this paper presents a hybrid human-machine pipeline. This pipeline begins with generating a small initial set of sub-questions using attribute-based templates, which are then refined through human annotation. This initial annotated set serves as the foundation for training a sub-question generator and a validator, and the generator and the validator together allow automatic construction of a large-scale dataset. As a result, this paper presents a new large-scale dataset, GQA-Q2Q, designed to disambiguate unclear entities through clarifying sub-questions. Furthermore, a VQA framework is introduced which utilizes the clarifying sub-questions to resolve entity ambiguity before producing a final answer. The experimental results demonstrate that this approach enhances VQA performance, validating the effectiveness of the proposed dataset.

1 INTRODUCTION

Visual question answering (VQA) aims to provide an accurate answer to a natural language question based on a given image related to the question. With the advancement of transformer-based vision-language models (VLMs), there have been remarkable performance improvements by the VQA models that adopt a VLM as a question-answering model across a variety of benchmarks (Bai et al., 2023; Liu et al., 2023). Despite the improvements, when a question mentions an unclearly specified entity, the VQA models often fail to derive an accurate answer from the image alone. For example, in Figure 1 sampled from GQA dataset (Hudson & Manning, 2019), the phrase ‘*happy man*’ in the question is originally intended to refer to the man on the right side of the image. However, since there is another happy man on the left side, there are two possible answers although only one of them is actually correct. Therefore, it is necessary to clarify such an ambiguous entity, and it is an effective and practical approach to introduce sub-questions for clarification.

In order to cope with the entity ambiguity in the image, this paper defines an ambiguous question as the one that lacks a clearly identifiable target entity for accurate reasoning within the context of VQA. Based on this definition, it presents GQA-Q2Q (*GQA-ambiguous Question to clarified Question*), a large-scale dataset of 135K sub-questions for the ambiguous questions of GQA dataset. Since GQA-Q2Q is constructed based on the GQA dataset, it allows direct utilization of the scene graph information for precise entity grounding. The sub-questions in this dataset are formulated as yes/no ones, and they are designed to clarify which entity instance the original question refers to, thereby resolving the entity ambiguity of the original question. By leveraging a sub-question and its sub-answer as well as the original ambiguous question, a VQA model can arrive at a correct answer even when the original question mentions an ambiguous entity. Therefore, the construction of a large-scale, high-quality dataset of clarifying sub-questions is essential for training and evaluating VQA models that can manage ambiguous entities.

054	
055	
056	
057	
058	
059	
060	
061	
062	
063	
064	Input
065	
066	Question: Is the happy man to the left or to the right of the baby?
067	Output
068	Gold answer: The man is to the right of the baby.
069	GPT4 answer: The happy man is to the left of the baby in the photo.
070	
071	
072	
073	
074	
075	
076	
077	
078	
079	
080	
081	
082	
083	
084	
085	
086	
087	
088	
089	
090	
091	
092	
093	
094	Despite the necessity of such a dataset, the manual construction of a large number of sub-questions is extremely expensive. Thus, this paper proposes a human-machine collaborative pipeline to construct a large, high-quality dataset of entity-clarifying sub-questions. The pipeline begins with manual construction of a small initial dataset by a human annotator. For every ambiguous question, a simple template-based sub-question generator prepares plural candidate sub-questions, and a human annotator chooses the best and plausible sub-question among the candidates if there is any. The automatic large-volume construction of sub-questions is performed in a similar way to the human construction. In this automatic construction, a VLM-based sub-question generator replaces the template-based sub-question generator, since the performance of the template-based generator is not reliable enough. In addition, the human annotator is replaced with a sub-question validator. These sub-question generator and validator are first trained with the initial dataset. The sub-question generator generates plural candidate sub-questions by nucleus sampling (Holtzman et al., 2020), and the validator selects only the highly confident candidates among the generated candidate sub-questions as human annotators do. Due to the large volume of ambiguous questions of GQA dataset and the high performance of the sub-question generator and validator, this human-machine collaboration achieves the scale and the quality of the proposed dataset simultaneously.
095	
096	
097	
098	
099	
100	
101	
102	
103	
104	
105	
106	
107	

Figure 1: An example ambiguous question in GQA dataset.

Despite the necessity of such a dataset, the manual construction of a large number of sub-questions is extremely expensive. Thus, this paper proposes a human-machine collaborative pipeline to construct a large, high-quality dataset of entity-clarifying sub-questions. The pipeline begins with manual construction of a small initial dataset by a human annotator. For every ambiguous question, a simple template-based sub-question generator prepares plural candidate sub-questions, and a human annotator chooses the best and plausible sub-question among the candidates if there is any. The automatic large-volume construction of sub-questions is performed in a similar way to the human construction. In this automatic construction, a VLM-based sub-question generator replaces the template-based sub-question generator, since the performance of the template-based generator is not reliable enough. In addition, the human annotator is replaced with a sub-question validator. These sub-question generator and validator are first trained with the initial dataset. The sub-question generator generates plural candidate sub-questions by nucleus sampling (Holtzman et al., 2020), and the validator selects only the highly confident candidates among the generated candidate sub-questions as human annotators do. Due to the large volume of ambiguous questions of GQA dataset and the high performance of the sub-question generator and validator, this human-machine collaboration achieves the scale and the quality of the proposed dataset simultaneously.

To validate the effectiveness of GQA-Q2Q, a VQA framework is introduced, which consists of an ambiguity detector, a sub-question generator, a sub-question respondent, and a final answerer. The ambiguity detector is a classifier that determines whether an input question contains an ambiguous entity. If the question is not ambiguous, the final answerer immediately generates a final answer with the input image and the question. Otherwise, the sub-question generator generates sub-questions to clarify the ambiguous entity, where it is a fine-tuned VLM with GQA-Q2Q. Then, the sub-question respondent (an oracle or a model that is aware of the ground-truth target entity) provides sub-answers to the sub-questions. Finally, a final answerer generates a final answer by leveraging the input image, the original question, the sub-questions, and the sub-answers.

The main contributions of this paper are as follows:

- This paper focuses on the entity ambiguity in VQA questions and constructs a high-quality dataset of 135,846 sub-questions from GQA dataset. To the best of our knowledge, this is the first large-scale dataset designed to resolve the entity ambiguity of VQA through targeted sub-question generation.
- This paper develops an efficient pipeline for dataset collection. The sub-question generator and the validator in the pipeline are trained with the initial dataset created by attribute-based templates and human verification, and then they are used to construct high-confidence sub-questions automatically. This enables a large and high-quality sub-question collection.
- This paper proposes a novel VQA framework that explicitly incorporates ambiguity resolution by answering a clarifying sub-question before answering the main question, which leads to improved performance of VQA.

108 **2 RELATED WORK**
109110 **2.1 VISUAL QUESTION-ANSWERING**
111112 Visual question-answering (VQA) is a representative multi-modal task in which a model answers
113 natural language questions about a given image. Many recent studies have introduced various VQA
114 datasets encompassing diverse types of visual information (Ren et al., 2015; Antol et al., 2015;
115 Goyal et al., 2017; Krishna et al., 2017; Marino et al., 2019; Wang et al., 2022). For instance,
116 GQA dataset provides scenario-based question-answer pairs derived from real images, where the
117 questions are generated from scene graphs and linguistic templates (Hudson & Manning, 2019).
118 VQA models also have evolved alongside the development of benchmark datasets. Recently, the
119 transformer-based models such as BLIP-2 (Li et al., 2023) and LLaVA (Liu et al., 2023) pretrained
120 with large-scale data have achieved strong zero-shot performances in various VQA benchmarks
121 (Goyal et al., 2017; Hudson & Manning, 2019; Marino et al., 2019).
122123 **2.2 QUESTION CLARIFICATION**124 Several studies have proposed question-clarification methods that resolve a question ambiguity in
125 VQA. These can be broadly divided into two categories. One category is to generate a clarified
126 version of an ambiguous question by removing ambiguity before passing it to a VQA model. Prasad
127 et al. (2024) proposed a method for clarifying ambiguous questions by rephrasing questions, aug-
128 menting them with visual groundings, and reasoning an answer with a confidence score. The other
129 category is to obtain additional information from a user via sub-questions, where the aim of sub-
130 questions is to clarify the intent of an original question (Selvaraju et al., 2020; Wang et al., 2022).
131 For example, Khan et al. (2023) improved zero-shot VQA performance by generating sub-questions
132 through question decomposition. However, most previous studies do not define clearly what consti-
133 tutes the ambiguity. Jian et al. (2025) proposed a benchmark called ClearVQA for clarifying ambigu-
134 ities in visual questions. While ClearVQA includes 9,243 ambiguous questions primarily addressing
135 text-based referential ambiguities, it remains relatively small in scale and does not explicitly focus
136 on entity-level ambiguities grounded in visual content.
137138 **2.3 AUTOMATIC DATA CONSTRUCTION**139 Automatic data construction is an effective alternative to costly manual annotation for large-scale
140 dataset creation for visual-language tasks. Some previous studies proposed large-scale datasets
141 constructed with hand-crafted templates (Ren et al., 2015; Hudson & Manning, 2019). However,
142 template-based data construction suffers from limited diversity and applicability, as it often produces
143 patterned data that may not fully reflect real-world scenarios. With the recent advances in generative
144 capability of LLMs that consider user instructions, data generation can be automated by using an
145 LLM (Liu et al., 2022; Wiegreffe et al., 2022). Lee et al. (2023) presented a large-scale dataset of
146 sensitive questions and acceptable responses, where the candidate questions and responses are gen-
147 erated by a machine and subsequently verified by human annotators. In this work, an initial dataset
148 is collected through human annotations, and a data generation model and an evaluation model are
149 trained with the annotated data to subsequently enable an automatic construction of a large-scale
150 dataset.
151152 **3 PRELIMINARIES**153 An ambiguous question in text-based question answering is defined as the one that has multiple pos-
154 sible answers or no definite answer at all (Min et al., 2020). While this definition is valid also for vi-
155 sual question answering, visual ambiguous questions require additional multi-modal comprehension
156 of images and texts. Prasad et al. (2024) proposed a method to handle ambiguous questions by refor-
157 mulating the questions into more specific and detailed ones based on visually grounded information.
158 However, the method has a limitation in that it just reformulates the questions, but does not resolve
159 the ambiguity within the questions explicitly. Chen et al. (2025) presented VQ-FocusAmbiguity, a
160 VQA dataset that grounds each question to its corresponding image region according to the concept
161 of focus ambiguity (Chen et al., 2025). However, VQ-FocusAmbiguity does not provide a concrete
162 approach for resolving the ambiguity.
163

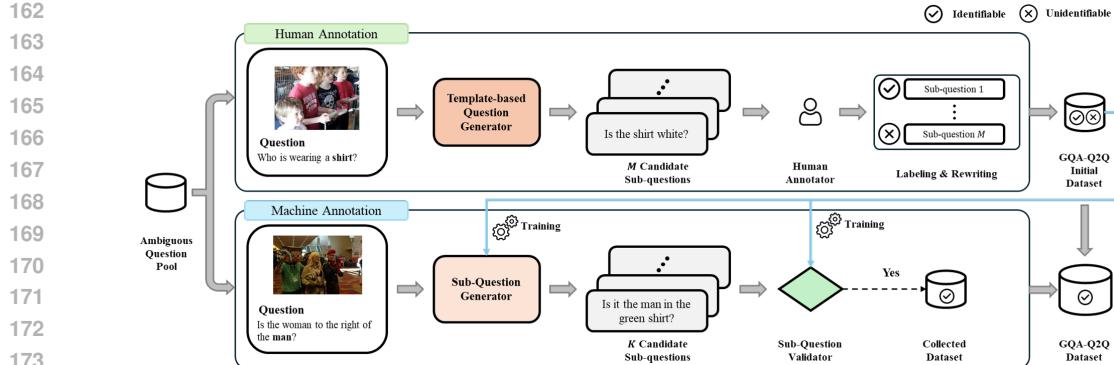


Figure 2: Overview of the GQA-Q2Q dataset construction process.

To effectively address ambiguous questions, it is essential to establish a clear criterion for determining whether a given question is ambiguous. In this paper, a question is defined as ambiguous if its entity corresponds to multiple instances within a given image. That is, a question is ambiguous if it exhibits *entity ambiguity*. Entity ambiguity is different from focus ambiguity in that it specifically addresses the problem of under-specified or ambiguous entities in VQA questions, while focus ambiguity refers more generally to situations that can be grounded to multiple regions of an image.

One possible solution for handling entity ambiguity is question clarification. This solution resolves ambiguity by asking a sub-question to an oracle or a questioner who is aware of the target entity. To eliminate ambiguity in a question, it is natural for humans to ask additional sub-questions that help clarify unclear parts of the original question. Thus, a good sub-question is the one that can uniquely identify the target instance in the image by referring to the distinguishable property of the entity. In this paper, the sub-questions are restricted to yes/no ones. Thus, receiving a ‘yes’ answer to a sub-question implies that the sub-question indeed refers to the target instance. For instance, a good sub-question to identify ‘*happy man*’, the ambiguous entity in Figure 1, could be “*Is the man wearing a hat?*” since only one among the two happy men wears a hat. After the clarification, a VQA model can answer the ambiguous question accurately by leveraging the sub-question and its sub-answer.

While sub-questioning is effective for resolving entity ambiguity in VQA, there is no available dataset designed for the benchmark of clarifying sub-questions. To fulfill this gap, this paper introduces GQA-Q2Q, a benchmark dataset for high-quality clarifying sub-questions. While manual construction of a dataset ensures high quality of the dataset, it is extremely expensive to construct it manually on a large scale. Recent advancements in large neural models showed that a machine trained with a small expert-annotated dataset can replace human experts effectively (Lee et al., 2023). Therefore, this paper employs a two-stage process for building GQA-Q2Q. The process begins with human annotation to develop a small initial dataset, and then enlarges the dataset with machine annotation. This leads to a dataset that achieves both scalability and high quality.

4 GQA-Q2Q DATASET

4.1 GQA DATASET AS A BASE DATASET

The proposed GQA-Q2Q dataset is based on GQA dataset, a widely used benchmark for visual question-answering. GQA dataset provides 113,018 images along with 22 million questions generated from scene graphs. Each scene graph contains information about an image including objects, attributes, and their relations. Moreover, since the target is annotated in each question, it is appropriate to identify the ambiguity of the entity of questions. This paper adopts the balanced version of GQA dataset, which contains 943,000 training, 132,062 validation, and 95,336 test questions, where the balanced version is obtained by downsampling unbalanced original dataset.

Recall that the questions that reference at least one entity appearing more than once in an image are ambiguous ones. Thus, a set of ambiguous questions, denoted as \mathcal{Q}_{amb} , is compiled from the

Dataset	Train	Validation	Test
# of sub-questions	58,646	7,297	7,337
– Identifiable	24,779	3,081	3,062
– Unidentifiable	33,867	4,216	4,275

Table 1: A simple statics on the initial GQA-Q2Q dataset.

Disambiguity		Fluency	
Human-annotated Data	Machine-annotated Data	Human-annotated Data	Machine-annotated Data
2.66 ± 0.18	2.42 ± 0.20	2.84 ± 0.17	2.80 ± 0.19

Table 2: Human evaluations of GQA-Q2Q in terms of disambiguity and fluency

training questions, and it has 125,854 (13.34%) ambiguous questions. The adjective modifiers in these ambiguous questions are removed to increase their ambiguity, where the modifiers are identified by a constituency parser ‘ptb-3-revised-electra-large’ from Stanza library¹. For example, the ambiguous question in Figure 1, “*Is the happy man to the left or to the right of the baby?*”, becomes more ambiguous if the modifier ‘*happy*’ is removed.

4.2 DATA COLLECTION OF GQA-Q2Q

Figure 2 illustrates how GQA-Q2Q dataset is constructed. The dataset is built in two stages. In the first stage, a small initial set is prepared through manual annotation. This initial set is used to train the sub-question generator and the validator for the next machine annotation stage. The sub-question generator generates a large scale of sub-questions from \mathcal{Q}_{amb} , and the validity of the generated sub-questions is ensured by the validator.

4.2.1 HUMAN ANNOTATION

For each question $q \in \mathcal{Q}_{\text{amb}}$, M candidate sub-questions are generated using five linguistic templates. The templates are designed to ask about a unique attribute of a target entity to distinguish the target instance from other instances. In Figure 2, the target entity is ‘*shirt*’ and one of the attribute values of the shirt is ‘*white*’. Thus, from the template “*Is <target entity> <attribute value>?*”, a candidate sub-question “*Is the shirt white?*” is generated. A more detailed explanation about the templates is provided in Appendix A.1.

The sub-questions that can identify a target instance are called *identifiable* sub-questions, while those that cannot identify a target instance are *unidentifiable* sub-questions. Note that the automatically-generated candidates contain both types of sub-questions. Thus, the candidates are validated by a human annotator. A human annotator reviews the candidates for q , and selects among them the most appropriate sub-question that allows the target instance to be identified. If a human annotator selects one appropriate sub-question, the remaining $M - 1$ candidates can be both identifiable and unidentifiable.

If none of the candidates are appropriate, the annotator creates a new identifiable sub-question manually. In this case, all generated candidates are definitely unidentifiable for q . These definitely unidentifiable sub-questions are also included in the initial set for the use of the next machine annotation. The number of sub-questions is summarized in Table 1. There are 73,280 sub-questions. Among them, 26,510 sub-questions are identifiable, and the remaining 46,770 sub-questions are unidentifiable.

4.2.2 MACHINE ANNOTATION

While human annotation ensures high data quality, it is costly and requires expert annotators with a thorough understanding of a task. To address these limitations and efficiently construct a large-scale GQA-Q2Q dataset beyond the initial set, an automatic data construction is employed. The automatic construction follows the process of the manual construction of the initial set, but the main problem with the manual construction is the reliability of the linguistic templates. Therefore, in the machine annotation stage, a sub-question generator and a sub-question validator are proposed instead of the templates and a human annotator to ensure the quality of automatically-generated sub-questions.

¹<https://stanfordnlp.github.io/stanza>

270
 271 Algorithm 1 describes the process
 272 of automatic construction of GQA-
 273 Q2Q. Since the initial set, \mathcal{D}_{init} , is
 274 available, it is used to train a sub-
 275 question generator $f_{SG}(\cdot)$ and a val-
 276 idator $f_{SV}(\cdot)$ by regarding identifiable
 277 sub-questions as positive samples and
 278 unidentifiable sub-questions as nega-
 279 tive samples. The sub-question gener-
 280 ator $f_{SG}(\cdot)$ is trained with only identi-
 281 fiable sub-questions, while the vali-
 282 dator $f_{SV}(\cdot)$ is trained with both identi-
 283 fiable and unidentifiable sub-questions.
 284 Recent large VLMs show strong per-
 285 formance on diverse tasks that re-
 286 quire both visual and textual reason-
 287 ing. Thus, LLaVA Liu et al. (2023) is
 288 adopted as a backbone model for im-
 289 plementing both $f_{SG}(\cdot)$ and $f_{SV}(\cdot)$.
 290

Algorithm 1 Automatic Data Construction for GQA-Q2Q

Require: \mathcal{D}_{init} (initial human-annotated dataset), \mathcal{D}_{id_init} (initial set of identifiable sub-questions), \mathcal{Q}_{amb} (ambiguous question pool), and τ (confidence threshold)

- 1: Train sub-question generator f_{SG} and validator f_{SV} using \mathcal{D}_{init}
- 2: $\mathcal{C} \leftarrow \{\}$ {Initialize new collection}
- 3: **for** each $q \in \mathcal{Q}_{amb}$ **do**
- 4: Obtain ambiguous entities E and image i for q
- 5: Generate K candidate sub-questions $\hat{Q} = \{\hat{q}_1, \dots, \hat{q}_K\}$ by
- $\hat{q}_k = f_{SG}(c_{SG}, i; \theta_{SG}), \quad k = 1, \dots, K$
- 6: **for** each $\hat{q}_k \in \hat{Q}$ **do**
- 7: Compute confidence score by
- $\langle s_k, p_{yes}^k \rangle = f_{SV}(c_{SV}, i; \theta_{SV}).$
- 8: **if** $p_{yes}^k \geq \tau$ **then**
- 9: Add (q, \hat{q}_k) to \mathcal{C}
- 10: **end if**
- 11: **end for**
- 12: **end for**
- 13: $\mathcal{D} \leftarrow \mathcal{D}_{id_init} \cup \mathcal{C}$
- 14: **return** final dataset \mathcal{D}

290 After that, sub-questions are generated for every ambiguous question using the sub-question gener-
 291 ator and the validator. That is, for each $q \in \mathcal{Q}_{amb}$, the sub-question generator first generates
 292 $\hat{Q} = \{\hat{q}_1, \dots, \hat{q}_K\}$, a set of candidate sub-questions. Assume that an ambiguous entity e in q has a set
 293 of instances $E = \{\bar{e}_1, \dots, \bar{e}_{|E|}\}$ in the image i , where only one instance is the ground-truth instance
 294 of \bar{e}^* . Then, $\hat{q}_k \in \hat{Q}$ is obtained by

$$\hat{q}_k = f_{SG}(c_{SG}, i; \theta_{SG}), \quad (1)$$

295 where c_{SG} is a text prompt that includes information on the bounding boxes of all instances in E
 296 and the highlighted target instance², and θ_{SG} is the fine-tuned parameters of LLaVA for sub-question
 297 generation. Since $f_{SG}(\cdot)$ is implemented by LLaVA, K sub-questions are generated by adopting the
 298 nucleus sampling following the work of Sultan et al. (2020).

301 Although all \hat{q}_i 's in \hat{Q} are generated by a VLM, some of them may not be identifiable. Therefore,
 302 the sub-question validator $f_{SV}(\cdot)$ filters out the unidentifiable sub-questions. For every $\hat{q}_k \in \hat{Q}$, its
 303 identifiability s_k and its confidence p_{yes}^k are computed by

$$\langle s_k, p_{yes}^k \rangle = f_{SV}(c_{SV}, i; \theta_{SV}). \quad (2)$$

306 Here, $s_k \in \{yes, no\}$ indicates whether \hat{q}_k is valid enough to identify the target instance. As in
 307 the sub-question generator, c_{SV} is a prompt that includes the bounding boxes of all instances of
 308 E and the highlighted target instance, and θ_{SV} is a set of LLaVA parameters for sub-question val-
 309 idation. This binary label s is derived from the predicted probability of the *yes* token, denoted as
 310 p_{yes} , obtained from the output distribution of $f_{SV}(\cdot)$. To ensure the high quality of sub-questions, a
 311 confidence-based filtering is applied with a threshold τ . That is, only the sub-questions with $p_{yes}^k \geq \tau$
 312 are included in \mathcal{C} , a set of automatically generated sub-questions. The number of sub-questions col-
 313 lected in this way is 109,336. When \mathcal{D}_{id_init} denotes a subset of \mathcal{D}_{init} of which members are all
 314 identifiable sub-questions, the final GQA-Q2Q dataset is a union of \mathcal{D}_{id_init} and \mathcal{C} . Thus, the total
 315 number of sub-questions in GQA-Q2Q is 135,846.

316
 317

4.3 DATASET ANALYSIS

318 For human evaluation of GQA-Q2Q, 150 sub-questions are randomly sampled from both human-
 319 annotated and machine-annotated data, respectively. Table 2 shows the results of human evaluations.
 320 Each sub-question is rated by three human evaluators on a three-point scale³ for disambiguity and
 321 fluency. The average scores for disambiguity are 2.66 for human-annotated sub-questions and 2.42

322
 323 ²Detailed explanation on prompts is given in Appendix A.2.

³This scale is further explained in Appendix A.3.

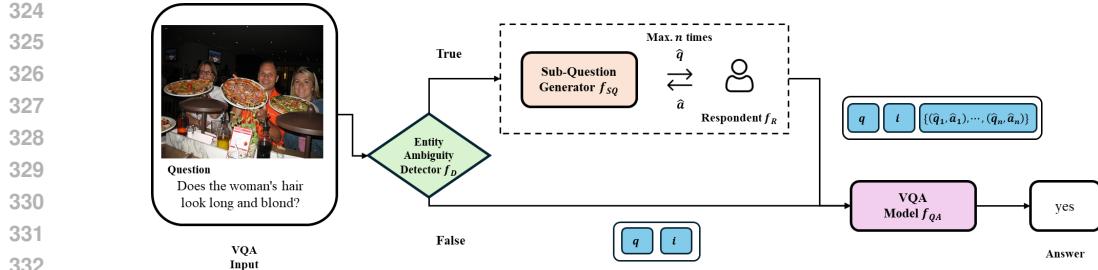


Figure 3: Overview of the proposed VQA framework for considering ambiguous questions.

for machine-annotated sub-questions. This result indicates that disambiguation is high for both types of sub-questions, and machine-annotated sub-questions are of similar quality to human-annotated ones. Fleiss' kappas and Krippendorff's alphas are over 0.4 and 0.6 for both types respectively, which implies moderate agreement among the evaluators. The average scores for fluency are also over 2.80 for both types. This result supports the linguistic plausibility of the sub-questions.

5 VISUAL QUESTION-ANSWERING WITH GQA-Q2Q

The validity of GQA-Q2Q dataset is shown by a VQA framework designed to leverage clarifying sub-questions to answer ambiguous questions. Figure 3 depicts the overall structure of the proposed VQA framework. Assume that a VQA sample $\langle q, i \rangle$ is given, where q is a question and i is an image related to q . The framework first applies the sample to the entity-ambiguity detector, $f_D(\cdot)$. For every entity e appearing in q , f_D determines the existence of its entity ambiguity by

$$y_{\text{amb}} = f_D(E),$$

where $y_{\text{amb}} \in \{\text{true}, \text{false}\}$ indicates the ambiguity existence of e and E is a set of instances of e in the image i . In the paper, $f_D(\cdot)$ is a simple count-based classifier which labels q as ambiguous if $|E| \geq 2$. That is, if any entity e has two or more instances, then q is ambiguous.

If q is not ambiguous, a VQA model directly predicts an answer to q using only q and i , since q does not contain any entity ambiguity. That is, the answer a of q is obtained by

$$a = f_{\text{QA}}(c_{qa}, i; \theta_{\text{QA}}), \quad (3)$$

where $f_{\text{QA}}(\cdot)$ is a VLM and c_{qa} is a prompt about q . If q is ambiguous, it is clarified by further asking sub-questions about the ambiguous entity in q . That is, for every entity instance $\bar{e}_k \in E$, the sub-question generator $f_{\text{SG}}(\cdot)$ re-trained with \mathcal{D} , GQA-Q2Q dataset, produces a clarifying sub-question \hat{q}_k about \bar{e}_k by Equation (1) for resolving the ambiguity of an instance e . Then, the respondent which is either an oracle or a model that is aware of the ground-truth target instance \bar{e}^* provides a sub-answer \hat{a}_k to \hat{q}_k . That is, when a respondent model, $f_R(\cdot)$, is a VLM, \hat{a} is obtained by

$$\hat{a}_k = f_R(c_{\text{resp}}, i; \theta_{\text{resp}}), \quad (4)$$

where c_{resp} is a prompt about \hat{q}_k , \bar{e}^* , and E , and θ_{resp} is the parameters for the VLM.

Note that \hat{a}_k is *yes* or *no* since \hat{q}_k is a yes/no question. It is *yes* if \hat{q}_k asks about the features of the ground-truth instance \bar{e}^* , and is *no* otherwise. This process of obtaining \hat{q}_k and \hat{a}_k is repeated up to n times to gather information about \bar{e}^* , where $n = |E|$. Thus, the collected pairs of a sub-question and its answer, $A = \{(\hat{q}_1, \hat{a}_1), \dots, (\hat{q}_n, \hat{a}_n)\}$, are used as an input to enhance the reasoning of the ambiguous question q .

Finally, $f_{\text{QA}}(\cdot)$, a VQA model implemented with a VLM, generates a final answer a from the ambiguous question q , the image i , the sub-question \hat{q} , and the set of sub-questions and sub-answers A by

$$a = f_{\text{QA}}(c_{qa}^{\text{amb}}, i; \theta_{\text{QA}}), \quad (5)$$

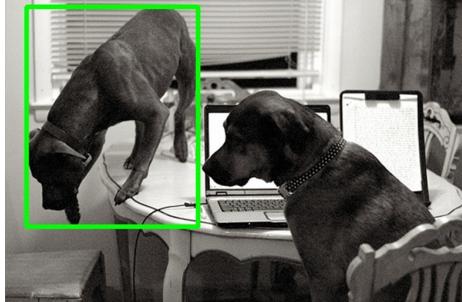
where c_{qa}^{amb} is a prompt explaining q and A , and θ_{QA} is the VLM parameters for question-answering. When compared with Equation (3), c_{qa} does not consider A since q is unambiguous in that case.

378
379
380

Sub-question Generator			Sub-question Validator	VQA Model	Respondent	
BLEU	ROUGE-L	BERTScore	Accuracy		Human Oracle	LLaVA
45.03	61.15	91.29	82.81%	LLaVA-1.5Vicuna 7B LLaVA-1.6Vicuna 7B	66.03% 70.15%	65.61% 69.49%

381
382
383
384

Table 3: The performances of the sub-question generator and the validator on the initial GQA-Q2Q test set.

385
386
387
388
389
390
391
392
393
394

Ambiguous Question: What device is to the right of the dog?
Sub-Question: Is the dog jumping off a table?

Human Annotation



Ambiguous Question: What does the man lean against?
Sub-Question: Is the man wearing a black jacket?

Machine Annotation

395
396
397

Figure 4: Two samples of GQA-Q2Q dataset from human annotation and machine annotation.

398
399
400

6 EXPERIMENTS

401
402
403
404

6.1 EXPERIMENTAL SETTINGS

405
406
407
408
409
410
411

LLaVA-1.6Vicuna 13B is adopted as a backbone model for both the sub-question generator and sub-question validator in machine annotation. In addition, the LoRA (Hu et al., 2022) is adopted to finetune their parameters efficiently. In the VQA task to show the effectiveness of GQA-Q2Q dataset, LLaVA-1.6Vicuna 7B is used for $f_R(\cdot)$ in Equation (4) if a VLM is used as a respondent model, and BLIP2 (Li et al., 2023), InstructBLIP (Dai et al., 2023), LLaVa-1.5 (Liu et al., 2023), and LLaVA-1.6 (Liu et al., 2024) are tested as a candidate backbone for $f_Q(\cdot)$ in Equation (5). All experiments below are performed on a machine with eight RTX 6000 Ada generation GPUs.

412
413
414
415
416
417
418

The hyper-parameters M and K in Figure 2 are both set to five, while τ to control the quality of sub-questions in Algorithm 1 is set to 0.9⁴. The high value of τ ensures that only the sub-questions that receive strong agreement from the validator are included in GQA-Q2Q. In addition, the maximum number of interactions between $f_{SG}(\cdot)$ and $f_R(\cdot)$ is limited to the number of entity instances. That is, $n = |E|$ is used in Figure 3. Automatic evaluation metrics of BLEU, ROUGE, and BERTScore are adopted to assess the sub-question generator, while the performances of the sub-question validator and VQA models are evaluated with accuracy. More settings are explained in Appendix A.5.

419
420

6.2 EVALUATION OF GQA-Q2Q

421
422
423
424
425
426
427
428

The initial GQA-Q2Q dataset is used to train the sub-question generator and the validator used during machine annotation. Table 3 shows their performance after trained with the training set of the initial GQA-Q2Q dataset, where the performance is measured on the test set of the initial GQA-Q2Q. The question generator achieves 45.03 of BLEU, 61.15 of ROUGE-L, and 91.29 of BERTScore, which proves that the sub-question generator is trustworthy. On the other hand, the accuracy of the validator is only 82.81%. This is the reason why even the sub-questions that receive a ‘yes’ answer from the validator should be filtered again using the hyper-parameter τ . Further details on the effect of the hyper-parameter K are provided in Appendix A.6.

429
430
431

Figure 4 shows two examples of the generated sub-questions. The left side of this figure comes from the initial set, where sub-questions are chosen by a human annotator. Here, the question mentions

⁴The reason why 0.9 is chosen is explained in Appendix A.4.

Method	Accuracy (%)
BLIP-2 _{Flan-T5 XL}	40.23
+ RepARe (Prasad et al., 2024)	46.36
+ Proposed Framework	42.56
+ (RepARe + Proposed Framework)	46.53
InstructBLIP _{Flan-T5 XL}	44.77
+ RepARe	49.67
+ Proposed Framework	48.69
+ (RepARe + Proposed Framework)	50.32
LLaVA-1.5 _{Vicuna 7B}	61.03
+ Proposed Framework	63.38
LLaVA-1.6 _{Vicuna 7B}	64.95
+ Proposed Framework	67.79

Table 5: VQA performances of the answering models with the proposed sub-question generator.

an ambiguous entity ‘*dog*’, but the sub-question distinguishes the target dog from the other dog by asking its attribute ‘*jumping off a table*’. Similarly, the right side is sampled from the machine-annotated sub-questions. This sub-question also identifies the target instance of the ambiguous entity ‘*man*’ by mentioning the attribute ‘*black jacket*’. More examples are given in Appendix A.7.

6.3 EVALUATION OF VQA WITH AMBIGUOUS QUESTIONS

Table 4 compares the VQA accuracies with a human respondent and LLaVA as a respondent. To evaluate them automatically, the identifiable sub-questions of \mathcal{D}_{init} are used instead of the sub-question generator in Figure 3, since the labels of the sub-questions in \mathcal{D}_{init} are already known. According to this table, the accuracies of LLaVA are close to those of a human respondent, regardless of which LLaVA version is used as a backbone answering model. This implies that LLaVA is as reliable as a human oracle and thus the proposed VQA framework can be completely automated by replacing the human respondent with LLaVA.

Table 5 proves that the proposed VQA framework is effective to enhance various VQA backbone models. All accuracies in this table are achieved by using LLaVA as a respondent model, and they are measured on the test set of the initial GQA-Q2Q. The proposed framework outperforms across all backbone models with improvements of 2.33% for BLIP2, 3.92% for InstructBLIP, 2.35% for LLaVA-1.5, and 2.84% for LLaVA-1.6, which indicates that the proposed VQA framework is effective in improving VQA performance on ambiguous questions, since the proposed generated sub-question and sub-answer resolve the entity ambiguity of ambiguous questions. Recall that RepARe rewrites ambiguous questions to unambiguous ones by considering various visual information. It also improves the accuracy of BLIP2 and InstructBLIP. However, RepARe and the proposed framework are not orthogonal. That is, even after RepARe rewrites an ambiguous question, the proposed framework can still resolve the entity ambiguity of the rewritten question. In this way, the accuracy of RepARe is improved for both BLIP2 and InstructBLIP. Nevertheless, the best accuracy of 67.79% is achieved when LLaVA-1.6_{Vicuna 7B} is used with the proposed framework. All these results demonstrate that the proposed framework is distinct and effective in resolving entity ambiguity.

7 CONCLUSION

This paper presented the first large-scale benchmark for resolving entity ambiguity in VQA through targeted sub-question generation. Based on the definition of ambiguous questions, GQA-Q2Q, a new dataset of clarifying sub-questions, was constructed from GQA benchmark. To achieve both scalability and high quality of the dataset, a human-machine collaborative pipeline was developed that combines template-based candidate generation, human verification, and model-driven data augmentation. In addition, a novel VQA framework was proposed to evaluate the validity of the proposed dataset. This framework explicitly incorporates ambiguity detection and resolution by utilizing the generated sub-questions and their corresponding answers before generating a final answer. The experimental results showed that leveraging the proposed GQA-Q2Q dataset leads to more accurate answers in scenarios involving entity ambiguity.

486 REFERENCES
487

488 Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zit-
489 nick, and Devi Parikh. Vqa: Visual question answering. In *2015 IEEE International Conference*
490 *on Computer Vision*, pp. 2425–2433, 2015.

491 Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang
492 Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, local-
493 ization, text reading, and beyond, 2023.

494 Chongyan Chen, Yu-Yun Tseng, Zhuoheng Li, Anush Venkatesh, and Danna Gurari. Acknowledg-
495 ing focus ambiguity in visual questions. In *Proceedings of the IEEE/CVF International Confer-*
496 *ence on Computer Vision*, pp. 1228–1238, 2025.

497

498 Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang,
499 Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: towards general-purpose vision-
500 language models with instruction tuning. In *Proceedings of the 37th International Conference*
501 *on Neural Information Processing Systems*, pp. 49250 – 49267, 2023.

502 Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in
503 vqa matter: Elevating the role of image understanding in visual question answering. In *2017 IEEE*
504 *Conference on Computer Vision and Pattern Recognition*, pp. 6325–6334, 2017.

505

506 Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and
507 Jeffrey P. Bigham. VizWiz Grand Challenge: Answering Visual Questions from Blind People. In
508 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617, 2018.

509 Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text
510 degeneration. In *International Conference on Learning Representations*, 2020.

511 Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,
512 and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International*
513 *Conference on Learning Representations*, 2022.

514

515 Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning
516 and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and*
517 *Pattern Recognition*, pp. 6693–6702, 2019.

518 Pu Jian, Donglei Yu, Wen Yang, Shuo Ren, and Jiajun Zhang. Teaching Vision-Language Models
519 to Ask: Resolving Ambiguity in Visual Questions. In *Proceedings of the 63rd Annual Meeting of*
520 *the Association for Computational Linguistics*, pp. 3619–3638, 2025.

521

522 Zaid Khan, Vijay Kumar BG, Samuel Schulter, Manmohan Chandraker, and Yun Fu. Exploring
523 Question Decomposition for Zero-Shot VQA. In *Proceedings of the 37th International Confer-*
524 *ence on Neural Information Processing Systems*, pp. 56615–56627, 2023.

525 Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie
526 Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei.
527 Visual genome: Connecting language and vision using crowdsourced dense image annotations.
528 *International Journal of Computer Vision*, 123(1):32–73, May 2017.

529

530 Hwaran Lee, Seokhee Hong, Joonsuk Park, Takyoung Kim, Meeyoung Cha, Yejin Choi, Byoung-
531 pil Kim, Gunhee Kim, Eun-Ju Lee, Yong Lim, Alice Oh, Sangchul Park, and Jung-Woo Ha.
532 SQuARE: A large-scale dataset of sensitive questions and acceptable responses created through
533 human-machine collaboration. In *Proceedings of the 61st Annual Meeting of the Association for*
Computational Linguistics, pp. 6692–6712, 2023.

534

535 Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image
536 pre-training with frozen image encoders and large language models. In *Proceedings of the 40th*
537 *International Conference on Machine Learning*, 2023.

538

539 Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. WANLI: Worker and AI col-
laboration for natural language inference dataset creation. In *Findings of the Association for*
Computational Linguistics: EMNLP 2022, pp. 6826–6847, 2022.

540 Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, volume 36, pp. 34892–34916, 2023.

541

542

543

544 Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee.

545 LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. <https://llava-vl.github.io/blog/2024-04-llava-next/>, 2024.

546

547 Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual

548 question answering benchmark requiring external knowledge. In *2019 IEEE/CVF Conference on*

549 *Computer Vision and Pattern Recognition*, pp. 3190–3199, 2019.

550

551 Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering

552 ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods*

553 *in Natural Language Processing*, pp. 5783–5797, November 2020.

554

555 Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Rephrase, Augment, Reason: Visual

556 Grounding of Questions for Vision-Language Models. In *Proceedings of the International Con-*

557 *ference on Learning Representations*, 2024.

558

559 Mengye Ren, Ryan Kiros, and Richard S. Zemel. Exploring models and data for image question an-

560 swering. In *Proceedings of the 29th International Conference on Neural Information Processing*

561 *Systems*, pp. 2953–2961, 2015.

562

563 Ramprasaath R. Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Tulio Ribeiro, Be-

564 smira Nushi, and Ece Kamar. Squinting at vqa models: Introspecting vqa models with sub-

565 questions. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.

566 10000–10008, 2020.

567

568 Md Arafat Sultan, Shubham Chadel, Ramón Fernandez Astudillo, and Vittorio Castelli. On the

569 importance of diversity in question generation for QA. In *Proceedings of the 58th Annual Meeting*

570 *of the Association for Computational Linguistics*, pp. 5651–5656, 2020.

571

572 Ruonan Wang, Yuxi Qian, Fangxiang Feng, Xiaojie Wang, and Huixing Jiang. Co-VQA : An-

573 swering by interactive sub question sequence. In *Findings of the Association for Computational*

574 *Linguistics: ACL 2022*, pp. 2396–2408, 2022.

575

576 Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. Reframing

577 human-AI collaboration for generating free-text explanations. In *Proceedings of the 2022 Con-*

578 *ference of the North American Chapter of the Association for Computational Linguistics: Human*

579 *Language Technologies*, pp. 632–658, 2022.

580

581 Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu,

582 and Lu Yuan. Florence-2: Advancing a Unified Representation for a Variety of Vision Tasks. In

583 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4818–4829, 2024.

584

585

586

587

588

589

590

591

592

593

594 **A APPENDIX**
595
596

Type	Template	Example
Attribute	Is/Are <target entity> <attribute value>?	<i>Is the shirt white?</i>
Relation	Is/Are <target entity> <relation value> <object>?	<i>Are the people located to the left of a pole?</i>
Position	Is/Are <target entity> on the left of the other <entity>? Is/Are <target entity> on the right of the other <entity>? Is/Are <target entity> in the middle of the other <entity>?	<i>Is the shirt on the left of the other shirts?</i> <i>Is the wine glass to the right of the other wine glasses?</i> <i>Is the man in the middle of the other men?</i>

602
603 Table 6: Template patterns for generating candidate sub-questions in human annotation and their
604 examples.
605
606607 **A.1 TEMPLATES FOR GENERATING CANDIDATE SUB-QUESTIONS IN HUMAN ANNOTATION**

608 The human annotation in Figure 2 is based on the candidate sub-questions generated automatically
609 with templates. The purpose of these sub-questions is to distinguish the target instance of an am-
610 biguous entity from other instances in the image. To achieve this, five templates for sub-question
611 generation are devised, where each template incorporates a unique property of the target instance.
612 Table 6 enumerates the templates and their examples. This paper considers three main types of
613 properties for the templates: attributes of an ambiguous entity, relations between entities, and their
614 spatial positions. The attribute-based sub-questions inquire about distinctive features of the am-
615 biguous entity, such as its color, and the relation-based sub-questions describe either spatial or semantic
616 relations between the ambiguous entity and other unambiguous entities. The final position-based
617 sub-questions localize the target instance of the ambiguous entity in relation to the relative locations
618 of other instances. In order to generate candidate sub-questions from the templates, the variables
619 in the templates are filled automatically by comparing the target instance with other instances or
620 the target entity with other entities in the scene graphs given in GQA dataset. Since the candidate
621 sub-questions are generated based on the scene graphs, they are informative and effective, enabling
622 a human annotator to efficiently select or refine high-quality sub-questions.
623

624 **A.2 PROMPTS FOR LLaVA**

625 This paper adopts LLaVA as a backbone model for the sub-question generator, the sub-question val-
626 idicator in machine annotation, and the respondent and the final answering model in VQA. LLaVA is a
627 VLM capable of generating a text response based on both a textual prompt and an input image. Each
628 module devises its own text prompt under the consideration of the instruction-following patterns
629 LLaVA was trained with in order to make the LLaVA perform appropriately in the targeted task.

630 Let $E = \{\bar{e}_1, \dots, \bar{e}_{|E|}\}$ be a set of entities appearing in the input image. The sub-question generator
631 aims to output identifiable candidate sub-questions that distinguish the target entity instance
632 among ambiguous instances. To generate such candidate sub-questions, spatial locations of the
633 entities represented by bounding boxes are essential for extracting the unique property of the target
634 entity. Thus, the prompt for the sub-questions generator, c_{SG} in Equation (1), is designed to be
635

< entity₁ >: $[x_1^{\bar{e}_1}, y_1^{\bar{e}_1}, x_2^{\bar{e}_1}, y_2^{\bar{e}_1}]$,
.....
< entity_{|E|} >: $[x_1^{\bar{e}_{|E|}}, y_1^{\bar{e}_{|E|}}, x_2^{\bar{e}_{|E|}}, y_2^{\bar{e}_{|E|}}]$

Target Entity: < entity_t >: $[x_1^{\bar{e}_t}, y_1^{\bar{e}_t}, x_2^{\bar{e}_t}, y_2^{\bar{e}_t}]$

Generate a sub-question to classify ambiguous entities.

Sub-Question:

636 where x and y are the coordinates of an entity and \bar{e}_t represented as < entity_t > is a target entity.
637 During dataset construction, \bar{e}_t is the ground-truth entity instance \bar{e}^* during dataset construction,
638 and is one of the entity instances in E in the VQA framework.
639

640 Similarly, the goal of sub-question validator is to determine whether a generated can-
641 didate can identify the target entity effectively. Therefore, its prompt is designed to in-
642 clude spatial locations of all instances of an ambiguous entity, the target instance, and
643 the sub-question. Consequently, when $I = \{\bar{e}_1, \dots, \bar{e}_{|I|}\}$ is a set of instances of an
644 ambiguous entity, the prompt for the sub-question validator, c_{SV} in Equation (2), is
645

648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702



Figure 5: An example of the prompt c_{SG} and c_{SV} and their corresponding outputs during machine annotation.

<entity₁>: [$x_1^{\bar{e}_1}, y_1^{\bar{e}_1}, x_2^{\bar{e}_1}, y_2^{\bar{e}_1}$],
 ...
 <entity_{|I|}>: [$x_1^{\bar{e}_{|I|}}, y_1^{\bar{e}_{|I|}}, x_2^{\bar{e}_{|I|}}, y_2^{\bar{e}_{|I|}}$]
 Target Entity: <entity_t>: [$x_1^{\bar{e}_t}, y_1^{\bar{e}_t}, x_2^{\bar{e}_t}, y_2^{\bar{e}_t}$]
 Sub-Question: <sub-question>
 Question: Does the sub-question classify the target entity?
 Answer:

where <sub-question> is a sub-question generated by the sub-question generator.

Figure 5 shows an actual usage of c_{SG} and c_{SV} . In this figure, the ambiguous entity is ‘man’, since there are two men in the image. Thus, the information about the coordinates of the two men and the target instance highlighted with a bounding box is included in both c_{SG} and c_{SV} . The difference between c_{SG} and c_{SV} is that a sub-question generated by the sub-question generator is provided at the end of the information in c_{SV} while c_{SG} has just an instruction. In this figure, the validator assigns the confidence score p_{yes} of 0.999 to the sub-question “Is the man wearing a black jacket?” generated by the sub-question generator. Since $p_{yes} > 0.9$, the sub-question is accepted as an identifiable sub-question.

The respondent in Figure 3 is a model that is aware of the ground-truth target instance. It outputs a sub-answer, either ‘yes’ or ‘no’, where ‘yes’ indicates the sub-question is correctly asking about the ground-truth target instance and the target instance has the property described in sub-question. Conversely, ‘no’ implies that the sub-question does not refer to the target instance or the target instance does not have the described

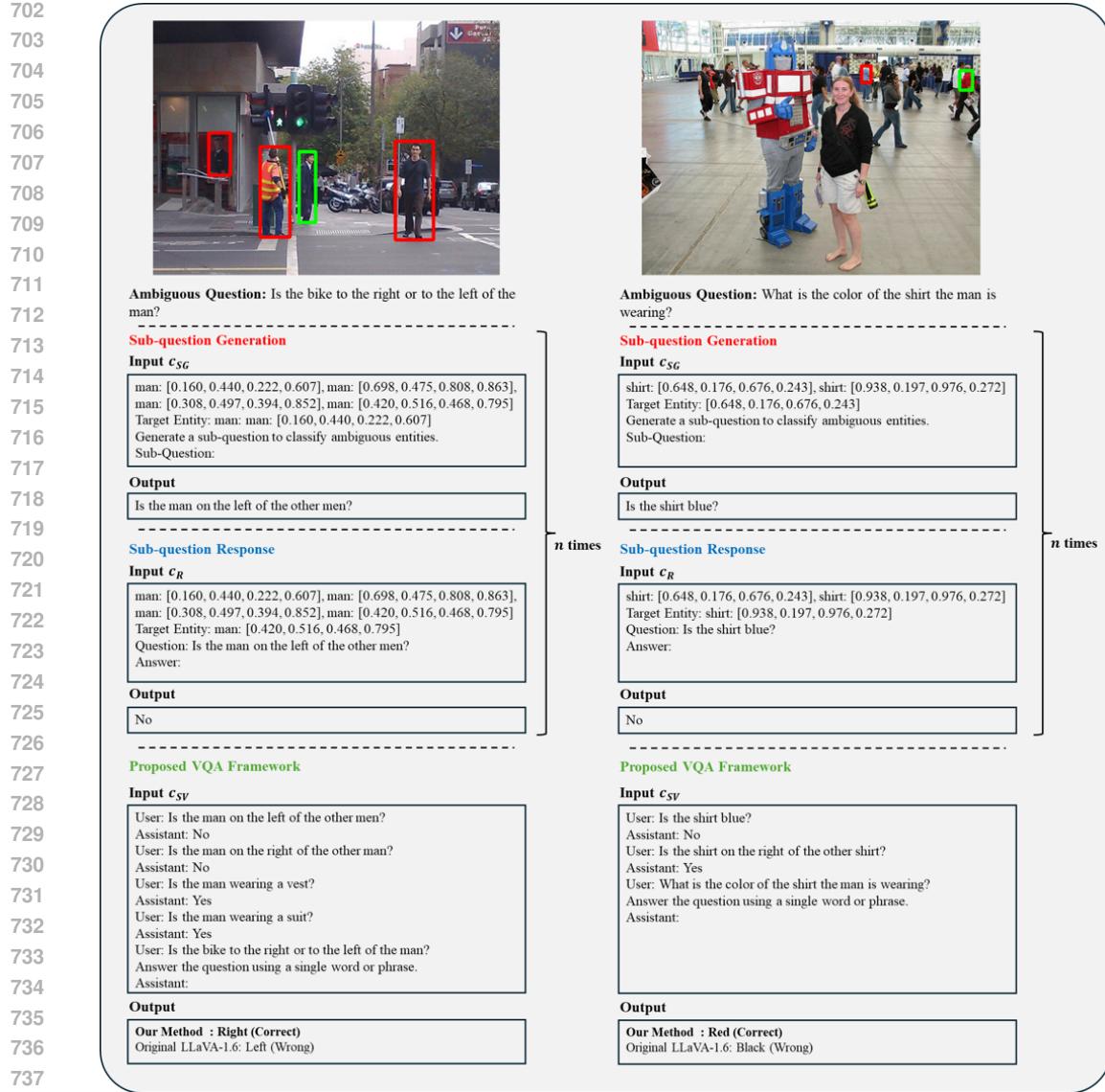


Figure 6: Two examples of the prompts c_{SG} , c_R , and c_{QA} and their corresponding outputs during VQA. In the images, the green bounding box indicates the ground-truth entity instance, while the red bounding boxes denote other ambiguous entity instances, which are visualized only for illustration and not provided as input to the model.

property. Therefore, the prompt for the respondent, c_R in Equation (3), is as follows.

```

<entity1>: [x1ē1, y1ē1, x2ē1, y2ē1],  

    ...  

<entity|E|>: [x1ē|E|, y1ē|E|, x2ē|E|, y2ē|E|]  

Target Entity: <entityt>: [x1ēt, y1ēt, x2ēt, y2ēt]  

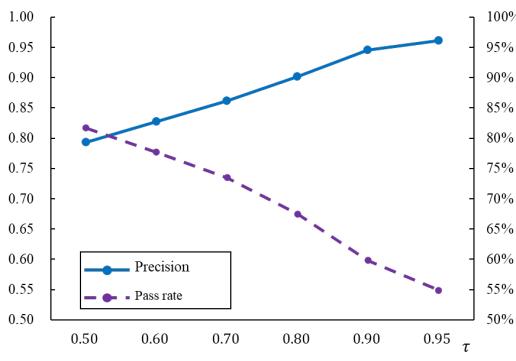
Question: <sub-question>  

Answer the question as Yes or No.

```

In Figure 5, the final answering model generates an answer based on an image, an ambiguous question, and n pairs of sub-questions and sub-answers. In the experiments, LLaVA-1.6Vicuna 7B achieves the highest accuracy among the backbone models such as the BLIP-2, InstructBLIP, and LLaVA-1.5. Following the instruction-tuning paradigm of LLaVA and chat templates from Vicuna,

756
757
758
759
760
761
762
763
764
765
766
767



768 Figure 7: The change of the precision and the retention rate of identifiable sub-questions according
769 to the values of τ .
770

771 the prompt c_{QA} is designed to guide LLaVA-1.6_{Vicuna 7B} to reason step-by-step using the context of

772 User: <sub-question 1>
773 Assistant: <sub-answer 1>
774 ...
775 User: <sub-question n >
776 Assistant: <sub-question n >
777 User: <question>
778 Answer the question using a single word or phrase.
779 Assistant:

780 where <question> is the ambiguous question and each pair of a sub-question and a sub-answer,
781 repeated n times, is generated by the sub-question generator and the respondent, respectively. By
782 structuring the prompt in this multi-turn dialogue format, the LLaVA-1.6 is encouraged to perform
783 a step-by-step reasoning before arriving at the final answer.

784 Figure 6 illustrates the entire process of the proposed VQA framework using the prompts c_{SG} , c_R ,
785 and c_{QA} . Three sub-questions are generated by the sub-question generator using c_{SG} , and they are
786 all determined as *identifiable* by the respondent using c_R . Thus, c_{QA} contains all these sub-questions
787 and sub-answers as well as the original ambiguous question. LLaVA-1.6 using c_{QA} gives a correct
788 answer, whereas the original LLaVA produces an incorrect answer.

789 A.3 HUMAN EVALUATION METRICS

790 To evaluate the sub-questions in the dataset qualitatively, a three-point scale test for disambiguity
791 and fluency is adopted. Disambiguity assesses how effectively a sub-question identifies the target
792 entity instance among ambiguous entity instances. A score of one implies that the sub-question does
793 not refer to the correct target entity at all, A score of two denotes that it is effective for multiple
794 instances including the target instance, and a score of three is assigned only when the sub-question
795 clearly distinguishes the target instance from others.

796 A.4 CONFIDENCE-BASED FILTERING

797 During machine annotation, a confidence-based filtering is used for the sub-question validator to
798 select only the high-quality sub-questions from candidate sub-questions. The accuracy of the sub-
799 question validator on the test split of the initial GQA-Q2Q dataset is 82.81% (see Table 3), but this
800 is not high enough to exclude all unidentifiable sub-questions. Thus, a confidence threshold τ is
801 applied to the result of the validator. That is, the sub-questions of which confidence is lower than τ
802 are excluded from the final GQA-Q2Q dataset even if they are determined to be identifiable by the
803 validator.

804 The precision and the pass rate are considered to find an optimal value of τ , where the precision
805 evaluates how accurately the validator identifies actual identifiable sub-questions and the pass rate
806 indicates how much percentage of the ambiguous questions have at least one sub-question that passes
807 this filter. With a high value for τ , no candidate sub-question of some ambiguous questions can pass

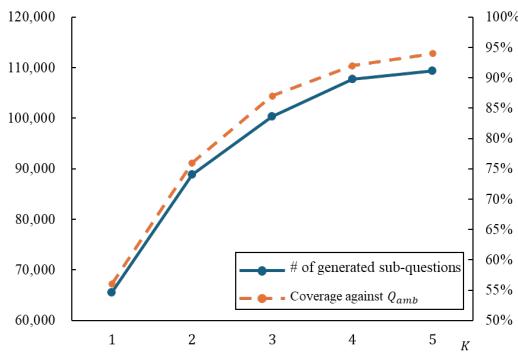


Figure 8: Change of the number of machine annotated sub-questions and their coverage against Q_{amb} according to K .

this filter, and then such ambiguous questions cannot be included in GQA-Q2Q. Figure 7 shows how the precision and the pass rate change according to τ , based on measurements taken from the test set of the initial GQA-Q2Q. The precision has a trade-off relation with the pass rate. Thus, the precision climbs up but the pass rate drops down, as τ increases. In addition, note that the gradient of the precision gets smaller from $\tau = 0.9$. Therefore, to achieve both the scalability and the quality of the final dataset, $\tau = 0.9$ is used in all experiments.

A.5 DETAILS OF EXPERIMENTAL SETTINGS

The sub-question generator and sub-question validator uses LLaVA-1.6_{Vicuna 13B} as a backbone VLM. They are fine-tuned on initial GQA-Q2Q dataset with five epochs, a batch size of 128, and a learning rate of 2e-4. To enable efficient parameter finetuning, LoRA is applied with a rank of $r = 128$ and ab alpha of $a = 256$. Adam optimizer is adopted to train them without weight decay and with a cosine learning with a warm-up ratio of 0.03. All trainings are conducted on a machine equipped with eight Ada generation GPUs of type RTX 6000, and two RTX 6000 Ada generation GPUs are used at the inference time of the sub-question generator and the validator. All source codes for training and inference are customized following the official source codes of LLaVA⁵ and RepARe⁶. The source codes and datasets will be publicly available after the reviewing process.

A.6 EFFECT OF K ON SUB-QUESTION GENERATION

Figure 8 depicts how the number of collected sub-questions is affected as the hyper-parameter K increases. The bigger K is, the more candidates the sub-question generator provides to the validator. Thus, as K increases, the more sub-questions are collected. However, the difference between $K = 4$ and $K = 5$ is small, which implies that the number of collected sub-questions would not increase though K is greater than five. Furthermore, when $K = 5$, sub-questions are generated from over 94% of the ambiguous questions in Q_{amb} . This is the reason why $K = 5$ is used in the experiments.

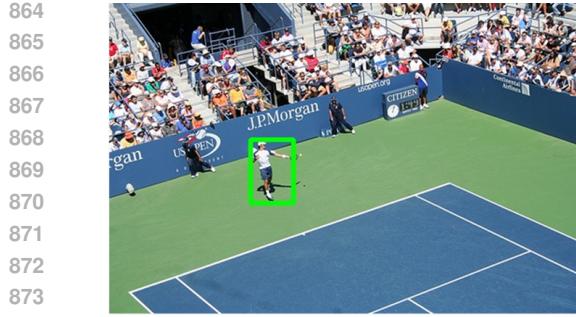
Fluency evaluates the linguistic quality of a text. It assesses whether the text is grammatically correct, coherent, and natural. A score of one for fluency means that a sub-question is ungrammatical or awkward. A score of two indicates that a sub-question understandable but slightly unnatural, while a score of three implies that a sub-question is fluent and well-formed.

A.7 ADDITIONAL EXAMPLES OF GQA-Q2Q

Following the samples shown in Figure 4, Figure 9 presents four additional samples of GQA-Q2Q, where two of them come from human annotation and the other two are sampled from machine annotation. In the human-annotated sample of the upper layer, the entity ‘man’ in the ambiguous question is unclear, since there are a number of men in the image. However, only one man is hitting

⁵<https://github.com/haotian-liu/LLaVA>

⁶<https://github.com/archiki/RepARe>



864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
13310
13311
13312
13313
13314
13315
13316
13317
13318
13319
13320
13321
13322
13323
13324
13325
13326
13327
13328
13329
13330
13331
13332
13333
13334
13335
13336
13337
13338
13339
13340
13341
13342
13343
13344
13345
13346
13347
13348
13349
13350
13351
13352
13353
13354
13355
13356
13357
13358
13359
13360
13361
13362
13363
13364
13365
13366
13367
13368
13369
13370
13371
13372
13373
13374
13375
13376
13377
13378
13379
13380
13381
13382
13383
13384
13385
13386
13387
13388
13389
13390
13391
13392
13393
13394
13395
13396
13397
13398
13399
133100
133101
133102
133103
133104
133105
133106
133107
133108
133109
133110
133111
133112
133113
133114
133115
133116
133117
133118
133119
133120
133121
133122
133123
133124
133125
133126
133127
133128
133129
133130
133131
133132
133133
133134
133135
133136
133137
133138
133139
133140
133141
133142
133143
133144
133145
133146
133147
133148
133149
133150
133151
133152
133153
133154
133155
133156
133157
133158
133159
133160
133161
133162
133163
133164
133165
133166
133167
133168
133169
133170
133171
133172
133173
133174
133175
133176
133177
133178
133179
133180
133181
133182
133183
133184
133185
133186
133187
133188
133189
133190
133191
133192
133193
133194
133195
133196
133197
133198
133199
133200
133201
133202
133203
133204
133205
133206
133207
133208
133209
133210
133211
133212
133213
133214
133215
133216
133217
133218
133219
133220
133221
133222
133223
133224
133225
133226
133227
133228
133229
133230
133231
133232
133233
133234
133235
133236
133237
133238
133239
133240
133241
133242
133243
133244
133245
133246
133247
133248
133249
133250
133251
133252
133253
133254
133255
133256
133257
133258
133259
133260
133261
133262
133263
133264
133265
133266
133267
133268
133269
133270
133271
133272
133273
133274
133275
133276
133277
133278
133279
133280
133281
133282
133283
133284
133285
133286
133287
133288
133289
133290
133291
133292
133293
133294
133295
133296
133297
133298
133299
1332100
1332101
1332102
1332103
1332104
1332105
1332106
1332107
1332108
1332109
1332110
1332111
1332112
1332113
1332114
1332115
1332116
1332117
1332118
1332119
1332120
1332121
1332122
1332123
1332124
1332125
1332126
1332127
1332128
1332129
1332130
1332131
1332132
1332133
1332134
1332135
1332136
1332137
1332138
1332139
1332140
1332141
1332142
1332143
1332144
1332145
1332146
1332147
1332148
1332149
1332150
1332151
1332152
1332153
1332154
1332155
1332156
1332157
1332158
1332159
1332160
1332161
1332162
1332163
1332164
1332165
1332166
1332167
1332168
1332169
1332170
1332171
1332172
1332173
1332174
1332175
1332176
1332177
1332178
1332179
1332180
1332181
1332182
1332183
1332184
1332185
1332186
1332187
1332188
1332189
1332190
1332191
1332192
1332193
1332194
1332195
1332196
1332197
1332198
1332199
1332200
1332201
1332202
1332203
1332204
1332205
1332206
1332207
1332208
1332209
1332210
1332211
1332212
1332213
1332214
1332215
1332216
1332217
1332218
1332219
1332220
1332221
1332222
1332223
1332224
1332225
1332226
1332227
1332228
1332229
13322210
13322211
13322212
13322213
13322214
13322215
13322216
13322217
13322218
13322219
13322220
13322221
13322222
13322223
13322224
13322225
13322226
13322227
13322228
13322229
13322230
13322231
13322232
13322233
13322234
13322235
13322236
13322237
13322238
13322239
13322240
13322241
13322242
13322243
13322244
13322245
13322246
13322247
13322248
13322249
13322250
13322251
13322252
13322253
13322254
13322255
13322256
13322257
13322258
13322259
13322260
13322261
13322262
13322263
13322264
13322265
13322266
13322267
13322268
13322269
13322270
13322271
13322272
13322273
13322274
13322275
13322276
13322277
13322278
13322279
13322280
13322281
13322282
13322283
13322284
13322285
13322286
13322287
13322288
13322289
13322290
13322291
13322292
13322293
13322294
13322295
13322296
13322297
13322298
13322299
133222100
133222101
133222102
133222103
133222104
133222105
133222106
133222107
133222108
133222109
133222110
133222111
133222112
133222113
133222114
133222115
133222116
133222117
133222118
133222119
133222120
133222121
133222122
133222123
133222124
133222125
133222126
133222127
133222128
133222129
133222130
133222131
133222132
133222133
133222134
133222135
133222136
133222137
133222138
133222139
133222140
133222141
133222142
133222143
133222144
133222145
133222146
133222147
133222148
133222149
133222150
133222151
133222152
133222153
133222154
133222155
133222156
133222157
133222158
133222159
133222160
133222161
133222162
133222163
133222164
133222165
133222166
133222167
133222168
133222169
133222170
133222171
133222172
133222173
133222174
133222175
133222176
133222177
133222178
133222179
133222180
133222181
133222182
133222183
133222184
133222185
133222186
133222187
133222188
133222189
133222190
133222191
133222192
133222193
133222194
133222195
133222196
133222197
133222198
133222199
133222200
133222201
133222202
133222203
133222204
133222205
133222206
133222207
133222208
133222209
133222210
133222211
133222212
133222213
133222214
133222215
133222216
133222217
133222218
133222219
133222220
133222221
133222222
133222223
133222224
133222225
133222226
133222227
133222228
133222229
133222230
133222231
133222232
133222233
133222234
133222235
133222236
133222237
133222238
133222239
133222240
133222241
133222242
133222243
133222244
133222245
133222246
133222247
133222248
133222249
133222250
133222251
133222252
133222253
133222254
133222255
133222256
133222257
133222258
133222259
133222260
133222261
133222262
133222263
133222264
133222265
133222266
133222267
133222268
133222269
133222270
133222271
133222272
133222273
133222274
133222275
133222276
133222277
133222278
133222279
133222280
133222281
133222282
133222283
133222284
133222285
133222286
133222287
133222288
133222289
133222290
133222291
133222292
133222293
133222294
133222295
133222296
133222297
133222298
133222299
1332222100
1332222101
1332222102
1332222103
1332222104
1332222105
1332222106
1332222107
1332222108
1332222109
1332222110
1332222111
1332222112
1332222113
1332222114
1332222115
1332222116
1332222117
1332222118
1332222119
1332222120
1332222121
1332222122
1332222123
1332222124
1332222125
1332222126
1332222127
1332222128
1332222129
1332222130
1332222131
1332222132
1332222133
1332222134
1332222135
1332222136
1332222137
1332222138
1332222139
1332222140
1332222141
1332222142
1332222143
1332222144
1332222145
1332222146
1332222147
1332222148
1332222149
1332222150
1332222151
1332222152
1332222153
1332222154
1332222155
1332222156
1332222157
1332222158
1332222159
1332222160
1332222161
1332222162
1332222163
1332222164
1332222165
1332222166
1332222167
1332222168
1332222169
1332222170
1332222171
1332222172
1332222173
1332222174
1332222175
1332222176
1332222177
1332222178
1332222179
1332222180
1332222181
1332222182
1332222183
1332222184
1332222185
1332222186
1332222187
1332222188
1332222189
1332222190
1332222191
1332222192
1332222193
1332222194
1332222195
1332222196
1332222197
1332222198
1332222199
1332222200
1332222201
1332222202
1332222203
1332222204
1332222205
1332222206
1332222207
1332222208
1332222209
1332222210
1332222211
1332222212
1332222213
1332222214
1332222215
1332222216
1332222217
1332222218
1332222219
1332222220
1332222221
1332222222
1332222223
1332222224
1332222225
1332222226
1332222227
1332222228
1332222229
1332222230
1332222231
1332222232
1332222233
1332222234
1332222235
1332222236
1332222237
1332222238
1332222239
1332222240
1332222241
1332222242
1332222243
1332222244
1332222245
1332222246
1332222247
1332222248
1332222249
1332222250
1332222251
1332222252
1332222253
1332222254
1332222255
1332222256
1332222257
1332222258
1332222259
1332222260
1332222261
1

918 These results indicate that entity-level ambiguity is a general phenomenon across VQA benchmarks.
 919 Although VizWiz exhibits a lower proportion of ambiguous questions compared to GQA and VQA-
 920 v2, this is expected: VizWiz consists of images captured by blind people, where questions often refer
 921 to a specific object or detail, resulting in fewer multi-instance ambiguities. In contrast, VQA-v2 con-
 922 tains a higher proportion of entity-ambiguous questions, demonstrating that GQA is not an outlier in
 923 terms of ambiguity frequency. GQA provides an appropriate and representative benchmark with the
 924 necessary grounding annotations to a large-volume of entities, making it suitable for constructing
 925 and evaluating our ambiguity-resolution pipeline.

926

927 A.9 APPLICABILITY TO OTHER VQA DATASETS

928

929 This paper develops and evaluates the ambiguity-resolution using GQA, a large-scale real-world
 930 VQA dataset whose scenegraphs and questions are generated through a fully automatic pipeline
 931 derived from Visual Genome annotations. This sub-question generation process demonstrates that
 932 entity-level grounding can produce algorithmically rather than manually, which is crucial for re-
 933 producibly identifying ambiguous instances.

934 The proposed pipeline is not restricted to GQA. Its applicability depends solely on the availability
 935 of entities referenced by the questions. Based on this requirement, the framework can be applied
 936 to other VQA datasets in two scenarios: (i) a dataset that contains grounding annotations, or (ii) a
 937 dataset without explicit grounding annotations. If grounding annotations are available, the pipeline
 938 can be applied directly. The required entities can be obtained automatically by generating scene
 939 graphs using grounding models trained on Visual Genome, as in GQA. If applying Visual Genome
 940 methods is challenging, a VLM model such as ‘Florence-2-Large’ (Xiao et al., 2024) can be used
 941 to extract entities from questions. Once a target entity is identified, the remainder of our framework
 942 operates unchanged. Overall, this paper proposes a dataset-agnostic pipeline that can be readily
 943 applied across a broad range of VQA benchmarks.

944
 945
 946
 947
 948
 949
 950
 951
 952
 953
 954
 955
 956
 957
 958
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968
 969
 970
 971