Why Did the Chicken Cross the Road?
Rephrasing and Analyzing Ambiguous Questions in VQA

Elias Stengel-Eskin **Jimena Guallar-Blasco** **Yi Zhou** **Benjamin Van Durme**
Johns Hopkins University
{elias, jgualla1, yzhou188@jhu.edu, vandurme}@jhu.edu

Abstract

Resolving ambiguities in questions is key to successfully answering them. Focusing on questions about images, we create a dataset of ambiguous examples; we annotate these examples, grouping the answers by the underlying question they address and rephrasing the question for each group to reduce ambiguity. An analysis of our data reveals a linguistically-aligned ontology of reasons for ambiguity in visual questions. We then develop an English question-generation model which we demonstrate via automatic and human evaluation produces less ambiguous questions. We further show that the question generation objective we use allows the model to integrate answer group information without any direct supervision.[1]

1 Introduction

The ability to ask questions allows people to efficiently fill knowledge gaps and convey requests; this makes questions a natural interface for interacting with digital agents. Visual question answering (VQA) models more specifically seek to answer questions about images, which can be useful in a variety of settings, such as assistive tech (Bigham et al., 2010). A number of datasets have been proposed for training VQA models, including VQAv2 (Goyal et al., 2017), VizWiz (Gurari et al., 2018), and GQA (Hudson and Manning, 2019). Such datasets are useful not only for training – they represent the aggregate judgements of speakers on a variety of factors, including ambiguity.

Ambiguity is a core feature of natural language, and can exist at all levels of linguistic analysis. In the context of data annotation, ambiguity often leads to disagreement between annotators. Given that the data resulting from crowdsourced annotation projects is typically used in a categorical fashion to train and evaluate models, annotator disagreements are problematic. Past work has often looked at detecting and resolving disagreements from the perspective of trust (Hovy et al., 2013), where some annotators are assumed to be more or less trustworthy. However, in the case of ambiguity, an annotator's honest effort might still lead to disagreement; in such cases, collecting more annotations can fail to establish a consensus. This differs from mistakes and cheating, where gathering more annotations would effectively outvote low-quality annotations. Ambiguity in the context of questions presents a particularly rich problem: firstly, question semantics are less clear from a formal point of view than the semantics of declarative sentences; this makes empirical accounts of questions particularly useful. Secondly, questions are increasingly relevant to natural language processing (NLP) research. Many NLP tasks are cast as question-answering (QA), including a growing number of tasks which can be cast as few-shot QA.

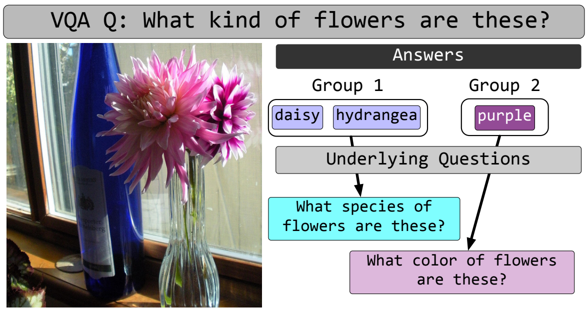

VQA Q: What kind of flowers are these?
Answers
Group 1
daisy
hydrangea
Group 2
purple
Underlying Questions
What species of flowers are these?
What color of flowers are these?

Figure 1: An ambiguous visual question from our dataset. Answers are grouped by the underlying question they answer, and the question is rephrased for each group. Answers within a group do not necessarily match, but do answer the same question.

Our main contributions are: (1) We examine how ambiguity appears in the VQAv2 data by constructing a dataset of 1,820 annotated visual image-question-answer triples. For each question, we ask annotators to re-group answers according to the underlying question they answer, and to rewrite ques-

[1] Code and data: https://github.com/esteng/ambiguous_vqa

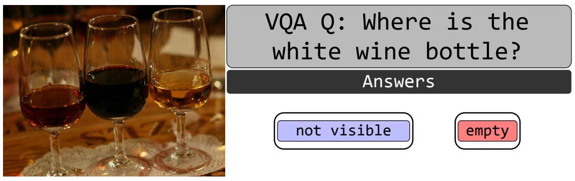


Figure 2: A visually underspecified question.

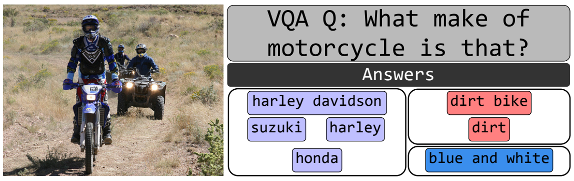


Figure 3: An underspecified and ambiguous question.

tions to unambiguously correspond to that group. (2) We create an ontology of causes for linguistic ambiguity based on the PropBank ontology (Kingsbury and Palmer, 2002; Gildea and Palmer, 2002; Palmer et al., 2005), and annotate our data with these causes. (3) We develop a visual question generation model which learns to rewrite questions; we validate this model with the re-grouped answers and re-written questions from our dataset. Our model can be used to cluster answers into their groups without any supervision for answer groups.

2 Ambiguity

In the VQAv2 annotations, each image has multiple questions, with each question being redundantly answered by up to 10 annotators. This redundancy is crucial for our annotations, as it provides us with multiple judgments per question, some of which may indicate ambiguity. We define ambiguous examples as ones where annotators are responding to different underlying questions. Note that this definition is not exhaustive, as it relies on the annotations; an example could be ambiguous but have few annotations, resulting in complete agreement between annotators. We contrast this definition with visual underspecification and uncertainty, which are categorized by a lack of visual information needed to answer a question, rather than ambiguity about what the question is. These can appear simultaneously, e.g. in Fig. 3 where there is both ambiguity and underspecification.

Fig. 2 gives an example of underspecification, as the information being queried is absent in the image and must be inferred. Past efforts examining reasons for annotator disagreement in VQA have addressed this distinction: Bhattacharya et al. (2019) introduce a dataset of 45,000 VQA examples annotated with reasons for disagreement, including ambiguity and lack of visual evidence as two separate categories. In practice, however, many examples labeled as ambiguous (such as Fig. 2) are cases of underspecification or unambiguous questions paired with visually ambiguous images. We use the ambiguous examples from Bhattacharya et al. (2019) as a starting point for our dataset.

3 Data

To properly study linguistic ambiguity in VQA, we collect a dataset of ambiguous examples, which represents a resource for categorizing and analyzing ambiguous questions and contains 1,820 answers to 241 image-question pairs. The data contains answers grouped by their underlying questions; there are 629 underlying questions.

The size of the ambiguous subset of VQA from Bhattacharya et al. (2019) prohibits our re-annotating the whole dataset, so we create a subset of data that is likely to be linguistically ambiguous. First, we sort the annotations into a priority queue using several heuristics. To merge synonymous answers (e.g. “cat”, “the cat”, “feline”) we embed each answer into continuous space using GloVe embeddings (Pennington et al., 2014), mean-pooling across words for multi-word answers and apply K-means (MacQueen, 1967; Lloyd, 1982) to the resulting embeddings, iteratively increasing the number of clusters k. Examples are scored by combining the K-means inertia score with a penalty for each additional cluster, trading off cluster coherence and having as few clusters as possible. These are subsequently sorted by how balanced their clusters are.[2] We remove yes-no questions with only “yes” and “no” answers, as they answer the same question. Note that we do not include questions from GQA (Hudson and Manning, 2019) in our dataset. Because GQA questions were generated from a grammar rather than being created by annotators, we do not expect there to be as much ambiguity. Furthermore, GQA questions are not included in the labeled data from Bhattacharya et al. (2019).

Annotation Interface We introduce a new annotation interface for re-grouping answers and re-writing questions (cf. Appendix C). We present the annotators with the question, image, and answers;

[2] Balanced clusters are more likely to be ambiguous, as unbalanced clusters are often a result of a single bad annotation.

Category	Property	PropB.	Description	Ex.
Property-based	Location	LOC	Asks about an object's location.	B.3.1
	Time	TMP	Asks about the time of an event or the time a picture was taken.	B.3.2
	Kind	N/A	Ask about what kind of something an object is.	B.3.3
Dynamic	Cause	CAU	Ask for the cause of an event.	B.4.1
	Purpose	PRP	Ask for the purpose of an event.	B.4.2
	Goal	GOL	Ask for the goal (location or person) of an object or event.	B.4.3
	Direction	DIR	Ask for the path being taken by an object.	B.4.3
	Manner	MNR	Ask in what manner an event is happening.	B.4.4
Pragmatic and Other	Multiple	N/A	Ask annotators to choose one of multiple options.	B.5.1
	Grouping	N/A	Ask annotators to group multiple items.	B.5.2
	Uncertainty	N/A	Contain visual uncertainty, especially for questions about events.	B.5.3
	Mistake	N/A	These involve bad answers or bad questions/images.	B.5.4

Table 1: Ontology of reasons why examples are ambiguous. Examples and details in Appendix B.

answers are pre-grouped based on the GLoVe K-means cluster assignments and are drag-able. Each answer cluster is paired with an editable text-box containing the original question. For each example, annotators have 3 tasks: first, they must decide whether the answers provided in the example correspond to different questions, or whether they all answer the same underlying question, i.e. whether the question is ambiguous or not. If an example is identified as being ambiguous, the second task is to re-group annotations by the question they answer. Each answer can be dragged into the appropriate cluster or deleted if it is spam; new clusters can also be created, and empty clusters can be deleted. Annotators were instructed to cluster answers by their underlying question, *not* by whether they are semantically similar. For example, antonyms like "good" and "bad" may be grouped into the same answer cluster. Finally, in the third task, annotators were asked to minimally edit the question corresponding to each created cluster, such that the new question uniquely corresponds to that cluster of answers. Instructions were presented to the annotators in text and video format. A local pilot with two vetted annotators was run to collect data for filtering annotators on Amazon MechanicalTurk (MTurk); only annotators with high agreement to the local annotators were allowed to participate in further annotation. Note that the local annotators were paid annotators unfamiliar with the goals of the project (i.e. not the authors). See Appendix B for details on the crowdsourcing process, including wage information. At least one author manually vetted all ambiguous examples, discarding noisy examples and editing questions for fluency. Examples were eliminated if the question could not be answered from the corresponding image (e.g. Fig. 2), or if the image had one or fewer viable responses. Edited questions were changed to improve the grammaticality of the rephrased questions; their content was left unedited.

Statistics Of the 1,249 examples run through MTurk, annotators skipped 942, identifying 307 as ambiguous. After cleaning these examples we have 241 unique image-question combinations, corresponding to 629 unique rewritten questions (including the examples from the pilot.) Each rewritten question is paired with 1-9 unique answers (mean: 2.9) – note that questions can have only one answer, since each example has multiple rewritten questions. We split our data into 30 dev questions and 211 test questions.

Inter-annotator Agreement We measure agreement on two levels: to what extent annotators identified the same examples as ambiguous, and the overlap between clusters of answers. Note that perfect inter-annotator agreement cannot be expected. Given that the examples we are interested in were ambiguous to the original set of VQAv2 annotators, with some seeing one reading over another, it is likely that some of the annotators in our task would also see only one reading.

Ambiguity agreement is defined as the percentage of examples two annotators both marked as being ambiguous. This number is averaged across annotator pairs. In the local pilot, the annotators had a pairwise ambiguity agreement score of 79.5%. In the MTurk pilot, 5 annotators had a mean pairwise score of 73.5% with a standard deviation of 6.0% (min 62.5%, max 80.0%). Note that we obtained redundant annotations only for the local and MTurk pilot HITs, and not the main data collection HIT.

The *cluster agreement* between two annotators is defined as the F1 score between the clusters of answers produced. Since the clusters are not aligned a priori, we use the Hungarian algorithm (Kuhn, 1955) to find a maximum overlap bipartite matching between clusters from each annotator and

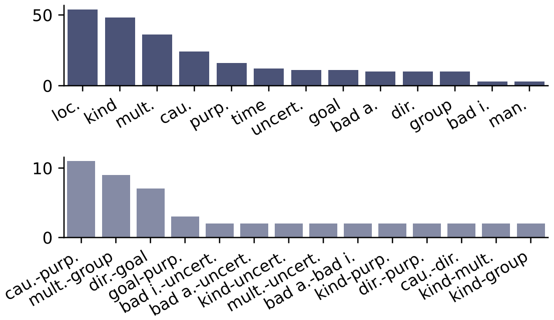


Figure 4: (Top) Frequency of each category. (Bottom) Co-occurrence frequency of each category (excluding frequencies ≤ 1). Some categories are highly correlated, indicating higher-order ambiguity.

then compute the F1 score between aligned clusters. These scores are averaged across annotator pairs. The local pilot cluster agreement score was 92.2, and the MTurk pilot's score was 88.4, with a standard deviation of 6.0 (min 77.1, max 94.6%).

Ambiguity Ontology After collecting the data, we observed that there were multiple groups within the ambiguous examples, corresponding to the factors that made a question ambiguous. We manually annotated all ambiguous examples according to the following linguistically-grounded ontology, which is largely aligned to PropBank roles (Kingsbury and Palmer, 2002; Gildea and Palmer, 2002; Palmer et al., 2005). The ontology is divided broadly into 3 categories. Property-based questions typically have to do with objects with multiple properties, and relate to partition question semantics (Groenendijk and Stokhof, 1984); more information can be found in Appendix B.1. Dynamic questions are about dynamic properties of objects or events. Finally, pragmatic ambiguities mainly relate to ambiguity in inferring the intention of the questioner, including choosing which element of the the world is most salient. Each category contains several subcategories – these are summarized in Table 1 and described in-depth in Appendix B.

Fig. 4 shows the frequency of each category, with the most common categories being location, kind, and multiple options, and shows the frequency with which pairs of categories co-occur (excluding pairs that only co-occur once). Several categories co-occur frequently, indicating higher-order ambiguity (i.e. ambiguity between what type of question is being asked). For example cause and purpose often co-occur; this indicates that they are often confused for each other, with some annotators providing answers consistent with a cause interpretation and others with a purpose interpretation. Furthermore, that they do not always co-occur indicates that ambiguity exists even within one interpretation.

4 Model

The data collected in Section 3 consists of questions rewritten according to their answer clusters. We develop a visual question generation (VQG) model which takes in answers and images and produces questions. After confirming the performance of the VQG model for generation generally, we evaluate the performance of a VQG model with respect to the answer clusters in our dataset. Specifically, we examine how the model can be used for clustering answers within an answer group together. Given that the answer clusters are based on the underlying question the answer is answering, we hypothesize that a good VQG model should not only learn to generate questions with a high similarity between the generated and reference questions, but learn input representations that contain answer group information. Note that this information would emerge in an unsupervised fashion, as we do not provide any answer group information during training.

We present a simple model for VQG consisting of a pre-trained vision-language encoder followed by a pretrained text-to-text encoder-decoder model. The encoder embeds an image and an answer into a shared representation space; the decoder produces a text question conditioned on this shared representation. We use ViLT (Kim et al., 2021) as our vision-language encoder. ViLT is a pre-trained fully transformer-based 87.4M-parameter model. The available ViLT model fine-tuned for VQA was trained on the entirety of the VQAv2 training data; since the annotations for Bhattacharya et al. (2019) come from the training set, our annotations also are sourced from the VQAv2 training set. To avoid test-set leakage, we fine-tune our own version of ViLT on a modified training set that excludes our annotations. Our input to ViLT is the image I_i and a text answer a_i from the set of answers for instance i, A_i. To generate text, we feed the output of ViLT to a pre-trained T5-base encoder-decoder model (Raffel et al., 2020) with $\sim$ 220M parameters, accessed via Huggingface Transformers (Wolf et al., 2020). We replace the T5 embedding layer with the output of our ViLT encoder, and train the model using all answers in the dataset with "yes" or "maybe" confidence ratings. We use categorical cross-entropy loss computed against the original question Q_i as our loss function. Note that the question Q_i is taken directly from the VQAv2 data, which we refer to as "original data" – we do not train on the annotations collected in Section 3.

Lexical Constraints Underspecification is a major challenge in VQG evaluation: given an image and an answer, there is often an intractably large set of questions that could have generated the answer. For example, in Fig. 1, the answer "purple" could also correspond to the question, "What color is the bottle's base?" Furthermore, even when the question is about the same topic, there are often a large number of semantically identical ways to phrase the question which may have very different surface forms. This poses a problem for surface-level evaluation metrics like BLEU. Finally, in our task of rephrasing questions, similarity is not a perfect predictor of quality. At one extreme, if the model generated the original question, it would receive a perfect similarity score when evaluated against the original question, but be as ambiguous as before. At the other extreme, as illustrated in the preceding example, a model may generate a valid question conditioned on the answer that has no relation to the original question's intent.

We attempt to tackle this problem by including positive lexical constraints from the original question in our decoding process. In a normal VQG setting, this would be impossible, since it requires the question at test time. However, in our setting, where the goal is to *rephrase* visual questions, we can assume access to questions. To generate a question on the same topic as the original, we use fast lexically-constrained decoding (Post and Vilar, 2018) with disjunctive positive constraints (Hu et al., 2019) during test decoding (+c in Table 2). We extract all contiguous noun spans from the question using Spacy's part-of-speech tagger (Honnibal and Montani, 2017); these are added as disjunctive positive beam search constraints so that the output contains at least one span. For example, without constraints, the question "Where are the people sitting?" (answer: "park") is rewritten "What kind of park is this?", while with constraints it would be "Where are the people?"

Baselines Due to the difference in our train and validation data as well as our use of constraints, our results are not directly comparable to previous VQG models. We instead compare our model to two baselines: "no image" (-v) and "no answer" (-t), where we give our model only the answer and only the image, respectively. These ablations verify our model's integration of multimodal information.

Training We use the VQAv2 training set for training, excluding the examples we annotated, which came from the train split. Since the answers for the VQA test split are not public, we use the validation data for testing and validation. We take $2,000$ questions pairs for validation and hold out the remaining $\sim 21K$ for testing. Each model was trained to convergence, measured by 5 consecutive epochs without BLEU score improvement, on four NVidia Quadro RTX 6000 GPUs; training took about 40 hours per model. All models were trained with the same hyperparameters (cf. Appendix D).

5 Visual Question Generation

Before analyzing performance on our dataset, we verify that the question-generation model we proposed is able to generate reasonable questions for the dataset more broadly. Here, we follow past work in reporting several string-based metrics: BLEU (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), Rouge-L (Lin, 2004) scores. We also report BertScore (Zhang et al., 2019).

Model	BLEU-4	CIDEr	ROUGE-L	BERT
iVQA*	0.21	1.71	0.47	N/A
VT5-v	0.22	1.51	0.45	0.93
VT5-v+c	0.21	1.82	0.47	0.93
VT5-t	0.16	1.00	0.32	0.92
VT5-t+c	0.18	1.51	0.38	0.92
VT5	0.27	1.98	0.48	0.94
VT5+c	0.26	2.21	0.50	0.94

Table 2: Test performance of the VQG model and baselines. Our model is able to integrate multimodal information and produce high-similarity questions.

Table 2 shows the test performance of the models tested, with and without constrained decoding. We see that the proposed generation model outperforms both baselines by a wide margin, indicating that it is successfully integrating information from both modalities. Furthermore, we see that in all cases, constraints improve performance; this is unsurprising, since the constraints force the model to include more of the reference question's n-grams. Finally, we include the performance of the iVQA model from Liu et al. (2018) in this table; however, we stress that the numbers are not directly comparable, since the training and evaluation data is different. Nevertheless, they help assert that our model is within the correct range for VQG.

Model as an Annotator In Section 3 we measured the inter-annotator agreement between annotators for clustering. We now compare the model predictions to these annotations with the same metric. Specifically, we measure how well the model's answer clusters align with annotated clusters, assuming access to the number of clusters given by the annotators. While this is a limiting assumption, it lets us evaluate to what degree the model's representations are useful in grouping answers, independently of whether the clustering algorithm can infer the right number of clusters. We hypothesize that the VQG loss will result in answer representations for answers to the same underlying question being more similar than answer representations for different underlying questions.

In order to obtain clusters from model representations, we use the K-means algorithm to group model representations of each answer $a_i \in A_i$. We then compare the F1 overlap between clusters produced by the model (and different clustering baseline) to the clusters produced by annotators using the method detailed in Section 3. We compare against several simple baselines. The **random** baseline randomly assigns answers to K clusters. The **perfect precision** baseline puts each answer in a separate cluster, leading to perfect precision but poor recall. The **perfect recall** baseline clusters all of the answers together, leading to perfect recall but poor precision. We also take the initial clustering of GloVe vectors with K-means, using an incrementally increasing K, as described in Section 3, as a baseline. For a more direct comparison, we extract the frozen pre-trained ViLT representation for the answer tokens and use mean pooling to combine them into a single vector per answer, clustering them with K-means for the **ViLT+K-means** baseline. Note that the ViLT representation is frozen and not trained for VQG. This baseline is contrasted with the **VT5 + K-means** system, where we extract mean-pooled answer token representations from the final layer of our VQG encoder and use these for clustering with K-means. Gains over the ViLT baseline reflect the benefits of the VQG loss combined with the T5 encoder pre-training.

Table 3 shows the clustering results. We see that VT5+K-means outperforms all baselines in F1, indicating that the representations learned via a VQG objective contain answer-group information. This is surprising, as the objective here does not directly optimize for answer groups; for a given training example (I_i, a_i, Q_i), there is a single reference output Q_i for all answers, regardless of the group they are in. However, the grouping information might be found in the dataset more broadly; when considering multiple examples with similar answers, answers in the same group may correspond to similar questions, leading them to be closer in representation space and thus in the same K-means cluster. In other words, the encoder representation for a given answer, having been trained across many similar questions and answers, is more similar within an answer group than across groups.

Method	Avg. P	Avg. R	Avg. F1
Human*	88.6	91.7	88.4
Random	64.9	70.4	59.4
Perfect P	100.0	50.6	61.1
Perfect R	63.4	100.0	76.3
GloVe initial	98.4	64.3	72.4
ViLT + K-means	65.9	68.6	60.1
VT5 + K-means	81.9	84.0	**79.0**

Table 3: Clustering metrics; Human results included for indirect comparison only.

6 Human Evaluation

The metrics in Section 5 suggest that our model holds promise as a method for rephrasing ambiguous questions; Table 2 indicates that the model produces fluent questions conditioned on images and answers, and Table 3 indicates that the model rewrites questions in a way that corresponds to the answer clusters and rewritten questions from human annotators. However, these automated metrics fall short of providing a full picture of the quality of rewritten questions, especially because, as mentioned before, it is not clear that similarity is a monotonic measure of success in our case. Thus, we conduct a human evaluation of 100 rewritten questions, specifically testing whether rephrased questions (from annotators and from the model) are less ambiguous than their original counterparts from the VQA dataset.

Methods Our evaluation paradigm presents annotators with an 3-way ordinal decision ("yes", "maybe", "no"), rating whether an answer is appropriate given an image and question. We sample 100 examples from our dataset; each example is paired with 3 questions: annotator-generated, model-generated, and original (from the VQAv2

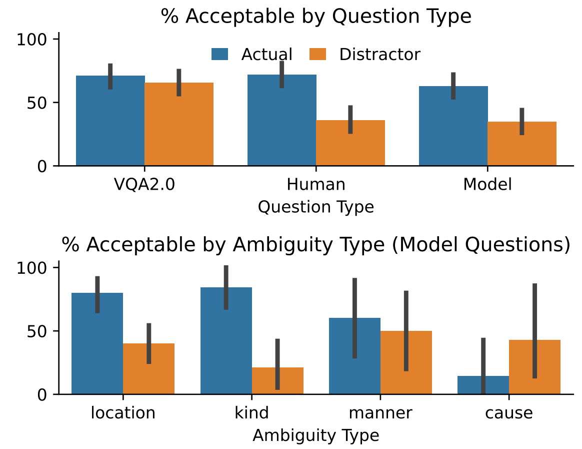


Figure 5: Percentage of answers rated as acceptable for each question type (annotator-rewritten, model-rewritten, original). Error bars represent bootstrapped 95% confidence intervals. Rewritten questions are less ambiguous (distractor rated as unacceptable) than their original counterparts. Model questions are generally less ambiguous across ambiguity categories.

dataset). The model-generated questions are taken from the VT5 model with constraints. For each image-question pair, we obtain 2 answers – one from the answer group corresponding to the rewritten question, and a distractor answer from a different answer group, as determined by the human annotations. In other words, for the example from Fig. 1, one non-distractor instance in the evaluation HIT would be the image, the question “What species of flowers are these?", and the answer “daisy”, while the distractor instance would have the answer “purple”. We would also have these two answers paired with the question “What kind of flowers are these?". An ambiguous question should be rated as acceptable for *both* answers (the actual and distractor), while a question rephrased to be less ambiguous should be rated as acceptable for the actual answer but *not* for the distractor answer, which corresponds to a different underlying question. Annotators were paid 0.04 per annotation for a total of 600 annotations, or $\sim$ $16 per hour.

Results and Analysis Fig. 5 shows the percentage of answers rated as acceptable (“yes” as opposed to “maybe” and “no”) across different conditions. The original, unedited question shows no significant difference between the actual and distractor answer, as measured by McNemar’s test (McNemar, 1947). This is expected, given that both answers (e.g. “daisy” and “purple”) were given by annotators in the original dataset to the original question, and thus are both likely to be viewed as acceptable. Both types of edited questions, on the other hand, show a significant difference between the actual answer and distractor answer, indicating that questions rephrased by annotators and by the model more specifically select answers from one answer group over, i.e. they are less ambiguous with respect to the answer group. The fact that the questions predicted by the model show only a small drop is promising, as it indicates that the model outputs are fluent and faithful to the original topic. Nevertheless, the model’s questions are rated as slightly less acceptable than the human questions, indicating room for improvement. In the bottom of Fig. 5 we see the percentage broken out by ambiguity type for the four most frequent types; here, we plot only the model-predicted sentences. We see that across most types there is a drop, with model outputs being rated as acceptable with the true answer, but not with the distractor.

7 Discussion

Limitations Our primary limitation is the size of our collected dataset; we have collected a quality dataset which we demonstrated is useful for analysis, but which is too small for training large-scale neural models. However, Section 5 indicates that a training-size dataset may not be necessary, as our question generation model is capable of capturing answer groups without explicit supervision. Another limitation on our dataset is the relative subjectivity of the task; in completing the annotation, we found that identifying ambiguity and isolating the different underlying questions often involves a Gestalt shift. Once an interpretation of the question is chosen, it becomes increasingly hard to see any other. This makes the annotation task subjective; where one annotator might see ambiguity leading to multiple valid answers, another might see one correct answer group and a number of invalid ones. Thus, the annotations in our dataset represent a high precision subset (rather than a high-recall subset) of all the possible ambiguous datapoints. We are also limited by the quality of the underlying data. Our dataset builds on the VQAv2 dataset (Goyal et al., 2017) and the annotations from Bhattacharya et al. (2019), both of which were large-scale annotation efforts intended for training. Due to their scale, individual datapoint quality is often quite low; this was one factor contributing to the need for post-hoc cleaning in the annotation process.

Future Work In addition to addressing these limitations, we leave exploiting the rewriting model to future work. In Table 2 and Fig. 5 we demonstrated that our question rephrasing model works well for producing fluent questions that reduce ambiguity. Furthermore, in Table 3 we showed that the model's representations contain information about the underlying question being asked, even though this information is not directly present in the training data and we do not include any supervision from our dataset. Future work could examine utilizing the rephrasing model in a search-engine environment, where users are actively querying about images. Given an ambiguous question identified and a set of answers to it from a VQA model, our model could be used to rephrase the question according to each answer. Just as a presenter will often rephrase a question from the audience, the model might present the user with the rephrased question it is actually answering, which would result in better interpretability. This improved interpretability might teach users how to interact with the model.

8 Related Work

Ambiguity Ambiguity in question-answering has been explored in the past: Min et al. (2020) introduce AmbigQA, a dataset of ambiguous open-domain questions paired with disambiguated rewrites. Our dataset differs in its domain: we address visual questions. Additionally, many of the ambiguities in AmbigQA are a result of background knowledge and changing dynamics. This is further explored by Zhang and Choi (2021), who introduce SituatedQA, a dataset of context-dependent questions and answers. In contrast, because VQA questions are closed-domain (i.e. they are typically about an image, not the world in general) the ambiguities we explore are more often a result of the language used in the question, rather than background knowledge of the annotator. Ambiguity has also been explored in natural language inference (NLI): Pavlick and Kwiatkowski (2019) explore annotator disagreement on NLI examples, finding ambiguity to be one source of disagreement.

Disagreement in VQA After the introduction of VQA datasets such as VQAv2 (Goyal et al., 2017) and VizWiz (Gurari et al., 2018), several papers focused on describing and diagnosing annotator disagreement in VQA. One line of work with deep ties to ours focuses on modeling annotator disagreement. Gurari and Grauman (2017) and Yang et al. (2018) present models for predicting annotator disagreement, which they use to reduce annotation cost. They both offer preliminary explorations of the features of high-disagreement questions. Bhattacharya et al. (2019) explore the reasons for disagreement in greater depth, annotating $\sim 45,000$ examples for the reason of disagreement; one of the possible reasons for disagreement is ambiguity. We use these in our collection (cf. Section 3). However, the data labelled as ambiguous in Bhattacharya et al. (2019) covers a range of phenomena, including visual ambiguity and underspecification, whereas our focus is specifically on linguistic ambiguity in visual questions.

Visual Question Generation Our work also relates to visual question generation (VQG). While VQG was first introduced as a task of generating unconstrained questions about images (Mora et al., 2016; Mostafazadeh et al., 2016), subsequent work has explored conditioning on images and answers to produce questions, as in Liu et al. (2018). Li et al. (2018) propose to generate questions as a dual auxiliary task for VQA, and Shah et al. (2019) use cycle consistency between generation and answering for improving VQA. Some past work has conditioned on partial answer information: Krishna et al. (2019) condition on answer categories rather than full answers, and Vedd et al. (2022) present a latent variable model which allows answers to be imputed at test-time. Terao et al. (2020) condition on answer-distribution entropy; in a similar vein to our work, Terao et al. focus on VQG for ambiguous questions. However, Terao et al. define ambiguity according to the entropy of their trained model and rely on user-specified entropy values for inference; we define it in a model agnostic way, according to features of the input. They also do not distinguish between linguistic and visual ambiguity.

9 Conclusion

We have presented a dataset of ambiguous VQA questions, annotated with reasons why they are ambiguous, as well as answers grouped by the underlying disambiguated question they are answering. We then introduced a model for rephrasing ambiguous questions according to their answers, finding that the model, which is trained purely on visual question generation, is able to recover information about the underlying question. We validate both our dataset and model using automatic and human evaluations, where we find that both reduce question ambiguity.

Acknowledgements

We would like to thank Nils Holzenberger and Kate Sanders for feedback on earlier drafts. Elias Stengel-Eskin is supported by an NSF GRFP.

References

Nuel D. Belnap and Thomas B. Steel. 1976. *The Logic of Questions and Answers*. New Haven/London: Yale University Press.

Nilavra Bhattacharya, Qing Li, and Danna Gurari. 2019. Why does a visual question have different answers? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4271–4280.

Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. 2010. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342.

Cassandra Chapman and Ivona Kučerová. 2016. Structural and semantic ambiguity of why-questions: An overlooked case of weak islands in english. *Proceedings of the Linguistic Society of America*, 1:15–1.

Donald Davidson. 1967. Truth and meaning. In *Philosophy, language, and artificial intelligence*, pages 93–111. Springer.

Daniel Gildea and Martha Palmer. 2002. The necessity of parsing for predicate argument recognition. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 239–246.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Jeroen Antonius Gerardus Groenendijk and Martin Johan Bastiaan Stokhof. 1984. *Studies on the Semantics of Questions and the Pragmatics of Answers*. Ph.D. thesis, Univ. Amsterdam.

Danna Gurari and Kristen Grauman. 2017. Crowdverge: Predicting if people will agree on the answer to a visual question. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3511–3522.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617.

C.L. Hamblin. 1958. Questions. *Australasian Journal of Philosophy*, 36(3):159–168.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with mace. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130.

J Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 839–850.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR.

Paul R Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *LREC*, pages 1989–1993.

Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. 2019. Information maximizing visual question generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2008–2018.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, and Ming Zhou. 2018. Visual question generation as dual task of visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6116–6124.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Feng Liu, Tao Xiang, Timothy M Hospedales, Wankou Yang, and Changyin Sun. 2018. ivqa: Inverse visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8611–8619.

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

J MacQueen. 1967. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *EMNLP*.

Issey Masuda Mora, Santiago Pascual de la Puente, and X Giro-i Nieto. 2016. Towards automatic generation of question answer pairs from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–2.

Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. *arXiv preprint arXiv:1603.06059*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. 2019. Cycle-consistency for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6649–6658.

Kento Terao, Toru Tamaki, Bisser Raytchev, Kazufumi Kaneda, and Shin'ichi Satoh. 2020. Rephrasing visual questions by specifying the entropy of the answer distribution. *IEICE Transactions on Information and Systems*, 103(11):2362–2370.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Nihir Vedd, Zixu Wang, Marek Rei, Yishu Miao, and Lucia Specia. 2022. Guiding visual question generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1640–1654, Seattle, United States. Association for Computational Linguistics.

Jette Viethen and Robert Dale. 2008. The use of spatial relations in referring expression generation. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 59–67.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Chun-Ju Yang, Kristen Grauman, and Danna Gurari. 2018. Visual question answer diversity. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*.

Michael Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Crowdsourcing

To collect a set of vetted data, a pilot task (or HIT) was run. A local annotator was paid $15 for one hour of annotation time (including watching the instruction video). The same annotations were then annotated by one of the authors. During this phase, the authors themselves ensured that there was no personally identifiable or offensive material in the data. From this data, we generated a set of examples for a pilot HIT to be run on Amazon's MechanicalTurk (MTurk).

To identify high-quality MTurk annotators, we ran pilot HIT of 41 examples from the local annotations, with 28 examples marked as ambiguous in the pilot and 13 examples marked as unambiguous (e.g. skipped). Workers were restricted to be located in the US. The annotations were presented sequentially, so that annotators had to complete all 41 examples to complete the HIT. Annotators were paid \$0.10 per example and received a 100% bonus for completing all examples (\$8 per HIT, roughly \$16 per hour of annotation).

From the pool of MTurk annotators who completed the pilot, we identified the top annotators. We then presented them with 850 examples in a non-sequential format, where each annotator could do as many as desired. No examples were flagged as offensive in this stage. Two annotators completed the task, which paid \$0.10 per example, with an \$8 bonus for every 300 examples. This corresponded to roughly \$16 per hour.

B VQA Ambiguity Ontology

B.1 Question Semantics

Formal semantics often focuses on variants of truth-conditional semantics, where knowing the meaning of an utterance is equated to knowing the conditions that would make the utterance true (Davidson, 1967). This account handles propositions well; however, evaluating the truth conditions of questions, an equally central feature of human language, seems more challenging. A rich literature has explored the meaning of questions (Hamblin, 1958; Belnap and Steel, 1976; Groenendijk and Stokhof, 1984, i.a.); for the purposes of this overview, we will briefly touch on one proposal which is of particular relevance to several categories outlined in Section 3. Under the partition semantics proposed by Groenendijk and Stokhof (1984), the meaning of a question is a set of utterances which partition the set of possible worlds. This is best illustrated with an example: assuming there were only two people in the whole universe ("John" and "Mary"), then the meaning of the question "Who walks?" is the partition introduced by the propositions "Only John walks", "Only Mary walks", "Both walk", "Neither walks". Each cell in the partition contains all possible worlds where the proposition is true, i.e. the "John walks" cell might contain a world where he walks outside, or on a treadmill, or one where the moon is made of cheese.

This proposal will describe a core feature of one type of disagreement we find. In certain cases, different answerers may have a different set of propositions in mind, leading to incompatible partitions. For example, given a picture of a blue children's tshirt, the question, "What kind of shirt is this" might be answered with "blue", "child's", or "small". In each of these cases, the partition function may be different, i.e. the "blue" answer is given as opposed to other colors, while the answer "child's" stands against "adult".

B.2 Property-based

Property-based ambiguities stem from annotators choosing to report different properties of objects or events with multiple properties. Another way to think of property-based ambiguities is in terms of the partition-based question semantics of Groenendijk and Stokhof (1984). Under partition semantics, the meaning of a question is a partition over possible worlds. These partitions can be described in terms of equivalence classes; for example, given a universe of two people ("John", "Mary") the partition induced by the question "Who walks?" has 4 cells, containing all worlds where only John walks, only Mary walks, both walk, and neither walk. In property-based ambiguities, annotators seem to choose different equivalence classes, which correspond to different cells in a partition and different sets of alternatives. For example, in Fig. 8, the annotator who says "white" is partitioning according to colors (e.g. "white sweater" as opposed to "blue sweater" or "black sweater") while the annotator who says "long sleeve" is partitioning possible worlds according sleeve style.

B.3 Property-based

There are three sub-classes of property-based ambiguities: location, kind, and time. (Back to table)

B.3.1 Location

Location maps to the PropBank tag `ARGM-LOC`. Answers here typically differ in terms of frame-of-reference, tracking with the observations of Viethen and Dale (2008).

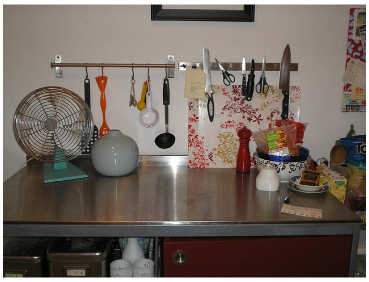


Figure 6: Question: Where is the fan? Answers: “on table”; “[l]eft side of counter in kitchen”

B.3.2 Time

This category maps to the PropBank tag ARGM-TMP. Answers often differ in terms of granularity and frame-of-reference (e.g. “morning”, “breakfast time”, “8am”).

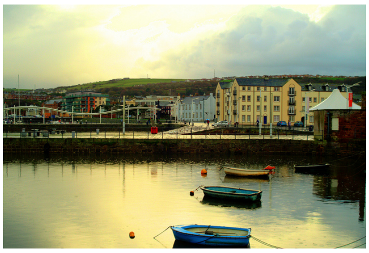


Figure 7: Question: What time of day is it? Answers: “morning”; “4 o’clock”

B.3.3 Kind

These do not map to PropBank, and ask about what type or kind of something an object is. Answers differ in terms of property class chosen.

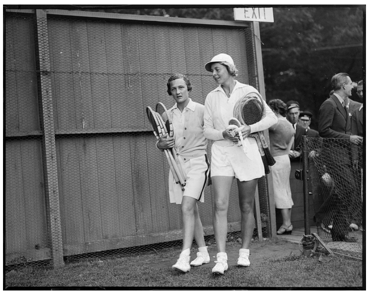


Figure 8: Question: What kind of top is she wearing? Answers: “white”; “button up to”; “sweater”; “long sleeve”

B.4 Dynamic

Dynamic questions are typically about properties of dynamic objects or events. Annotators often disagree on the type of question being asked (e.g. *cause* vs. *purpose*), as well as the underlying question within a type. These questions commonly correspond to “why” and “how” questions. (Back to table)

B.4.1 Cause

Maps to ARGM-CAU. These ask for the cause of an event. Since cause and purpose are often ambiguous (Chapman and Kučerová, 2016) annotators may differ here, and since cause is often underspecified from a static image, annotators may impute different causes. Even when causes are not imputed, annotators often may choose one of multiple causes, or report causes at different levels of granularity.

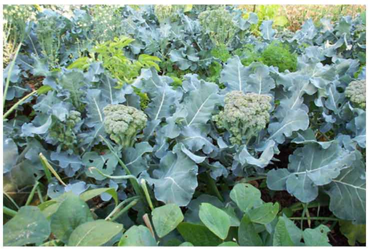


Figure 9: Question: Why is this blue and green? Answers: “it’s vegetables”; “cold”; “photosynthesis”; “garden”

B.4.2 Purpose

maps to ARGM-PRP. Purpose questions ask for the purpose of an event, and share their features with the *cause* examples.

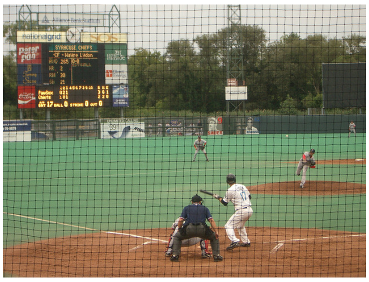


Figure 10: Question: What is the netting for? Answers: “baseball”; “ball”; “protect public”; “protect spectators”; “safety”; “don’t get hit by ball”;

B.4.3 Goal and Direction

Goal maps to ARGM-GOL and asks for the eventual goal (location or person) of an object or event. When the goal is a person, it is often the person who benefits from an action. Goals are often imputed, and can often be ambiguous with direction. Direction maps to ARGM-DIR and asks for the path being taken by an object. This is often ambiguous with goal, and is also often imputed or dependent on the frame-of-reference.

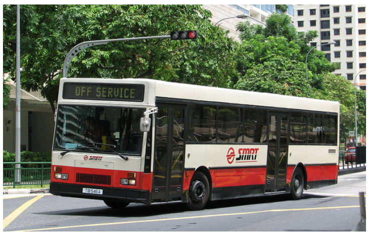


Figure 11: Question: Where is the bus going? Answers: "station"; "around corner"

B.4.4 Manner

Manner maps to `ARGM-MNR` and asks in what manner an event is happening. Manner questions can be ambiguous with cause questions.

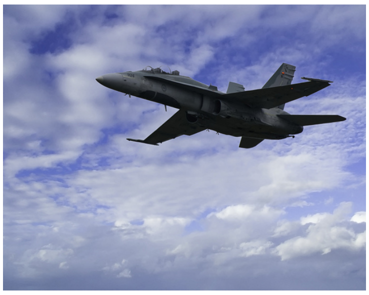


Figure 12: Question: How is the plane flying? Answers: "low"; "engines"; "in air"

B.5 Pragmatic/Other

Pragmatic ambiguities are typically characterized by an underspecified question which requires the answerer to infer a preference on the part of the questioner. For example, in the "Multiple Options" ambiguity, there are several valid responses, and different answerers might infer that different options are more or less salient to the questioner. None of the pragmatic ambiguities are aligned with PropBank. (Back to table)

B.5.1 Multiple Options

A common source of disagreement is when annotators are asked to choose one of multiple options. For example, a question like "what color is X?" when X has multiple colors will often result in a variety of answers. Here, the ambiguity is with respect to the inferred intent of the questioner; the answerer must infer which option is most salient to the questioner.

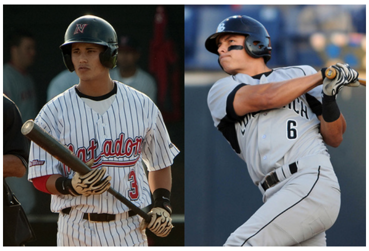


Figure 13: Multiple options ambiguity example. Question: What team is the man holding the bat playing for? Answers: "matadors"; "yankees"

B.5.2 Grouping

Grouping ambiguity often co-occurs with multiple options, and involves grouping several options; different annotators may include or exclude items from their groups.

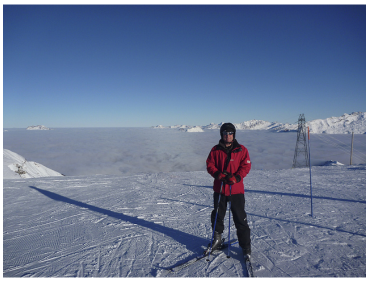


Figure 14: Question: What is on the right of the picture? Answers: "sky posts"; "mountain"; "electric tower, ski pole, and mountain top";

B.5.3 Uncertainty

Many examples contain visual uncertainty, especially for questions about events, which are inherently hard to capture in a static image.

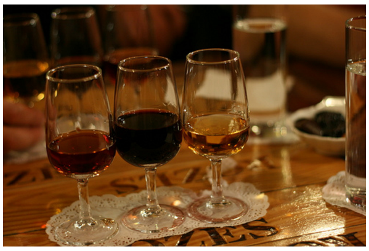


Figure 15: Uncertainty example. Question: Where is the white wine bottle? Answers: "not visible"; "empty"

B.5.4 Annotator mistakes

Some annotators provide bad or unreasonable answers to questions.

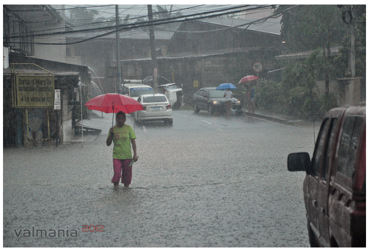


Figure 16: Annotator mistake. Question: How high is the water? Answers: “2-3 inches”; “rain water”

B.5.5 Bad question/bad image

Some questions are nonsensical and some images are extremely low quality, making answering any question about them impossible.

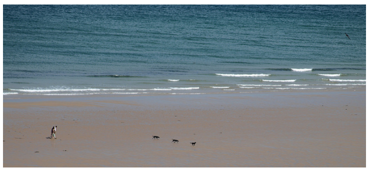


Figure 17: Bad image or data. Question: Which bird looks about to take off the ground? Answers: “middle bird”; “left 1”

C Interface

Fig. 18 shows the annotation interface used to collect the dataset. Answers are drag-able objects and can be moved across columns. New answer groups can be added. Questions are auto-populated with the original question and then edited by the annotator. Skipping opens up a text box with an auto-populated reason (“All answers to the same question”) that can be edited.

D Hyperparameters

Models were trained with the AdamW optimizer (Loshchilov and Hutter, 2018) using a learn rate of $1e-4$ with linear weight decay of 0.01. The learn rate followed a linear warmup schedule with $4,000$ warmup steps. The batch size was set to 32 per GPU, leading to an effective batch size of 128. As fine-tuning ViLT for VQG had no substantial impact, we freeze the ViLT encoder during training.

E Validation Performance

Table 4 shows the validation performance for all metrics reported in Table 2. Trends mirror those seen in the test data.

F Why Ambiguity

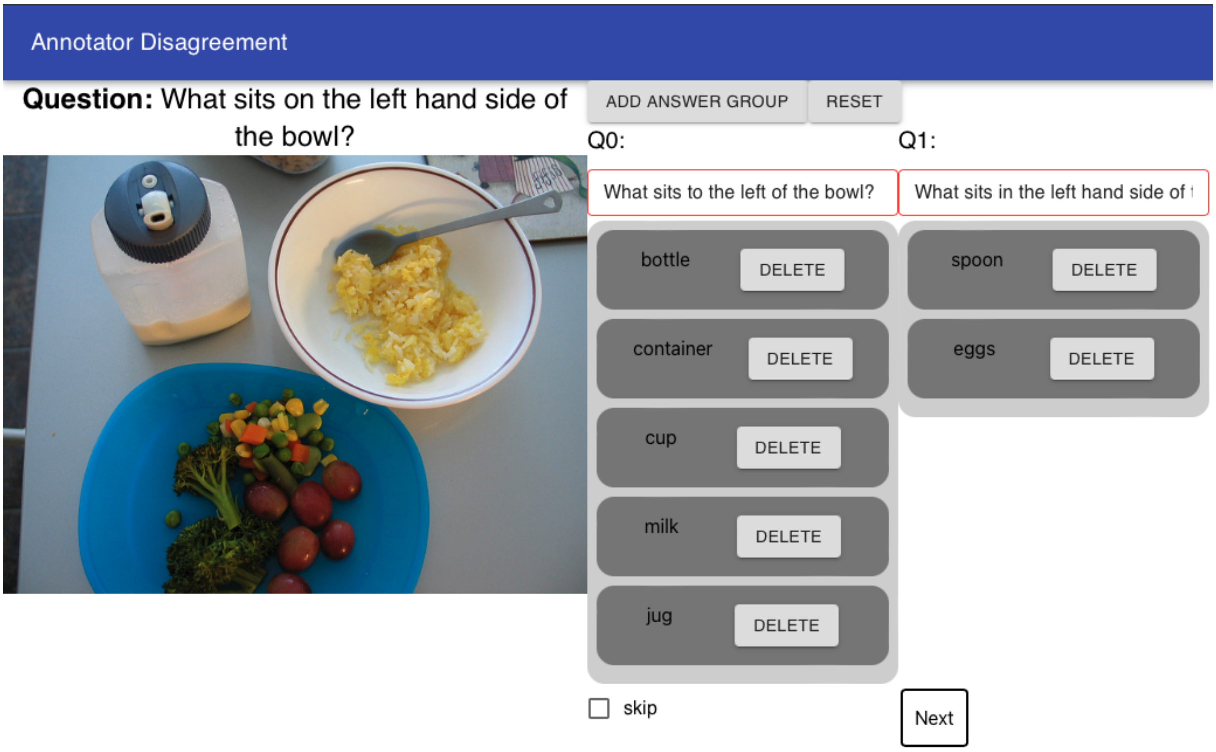
Fig. 19 gives an example of each possible combination of dynamicity and agency. Answers are grouped by whether they are cause or purpose answers. Note that grouping these answers is an underspecified task – in many cases, both readings can be coerced using context.

	–DYNAMIC	**+DYNAMIC**
+AGENTIVE	Why are these people on their cell phones **Purpose**: talk to people…, taking calls, talking, talking to someone, posing **Reason**: boredom, poor real life social skills	Why is the girl holding an umbrella? **Purpose**: keep sun off, block sun, sunscreen, sun protection, protection from sun **Reason**: it’s sunny, sun
–AGENTIVE	Why does the horse have a blanket on? **Purpose**: to warm its body, warmth **Reason**: cold, it is cold	Why is the man mid–air? **Purpose**: tricks, making ski jump, jumping **Reason**: snowboarding, he went off jump, snowboarding down hill, snowboarding jump, ramp

Figure 19: Example of an ambiguous annotation for each category in our “why” question analysis

G License

Code and data will be released under an MIT license.



Question: What sits on the left hand side of the bowl?

Q0: What sits to the left of the bowl?

- bottle
- container
- cup
- milk
- jug

Q1: What sits in the left hand side of t

- spoon
- eggs

Figure 18: The annotation interface.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	CIDEr	ROUGE-L	METEOR	BERT
iVQA*	0.43	0.33	0.26	0.21	1.71	0.47	0.21	N/A
VT5-v	0.47	0.31	0.22	0.16	1.05	0.42	0.41	0.93
VT5-t	0.39	0.21	0.14	0.10	0.48	0.29	0.30	0.91
VT5	0.53	0.37	0.28	0.22	1.51	0.46	0.47	0.94
VT5-v+c	0.47	0.30	0.21	0.15	1.33	0.43	0.45	0.93
VT5-t+c	0.42	0.25	0.17	0.12	0.95	0.34	0.38	0.92
VT5+c	0.53	0.37	0.27	0.21	1.73	0.47	0.50	0.94

Table 4: Validation performance of the VQG model and baselines. Our model is able to integrate visual and textual information and output questions with high similarity to reference questions.