CAISE: Conversational Agent for Image Search and Editing

Hyounghun Kim[1], Doo Soon Kim[2], Seunghyun Yoon[3]
Franck Dernoncourt[3], Trung Bui[3], Mohit Bansal[1]

[1]UNC Chapel Hill [2]Roku Inc. [3]Adobe Research
{hyounghk, mbansal}@cs.unc.edu
{syoon, dernonco, bui}@adobe.com

Abstract

Demand for image editing has been increasing as users' desire for expression is also increasing. However, for most users, image editing tools are not easy to use since the tools require certain expertise in photo effects and have complex interfaces. Hence, users might need someone to help edit their images, but having a personal dedicated human assistant for every user is impossible to scale. For that reason, an automated assistant system for image editing is desirable. Additionally, users want more image sources for diverse image editing works, and integrating an image search functionality into the editing tool is a potential remedy for this demand. Thus, we propose a dataset of an automated **C**onversational **A**gent for **I**mage **S**earch and **E**diting (CAISE). To our knowledge, this is the first dataset that provides conversational image search and editing annotations, where the agent holds a grounded conversation with users and helps them to search and edit images according to their requests. To build such a system, we first collect image search and editing conversations between pairs of annotators. The assistant-annotators are equipped with a customized image search and editing tool to address the requests from the user-annotators. The functions that the assistant-annotators conduct with the tool are recorded as executable commands, allowing the trained system to be useful for real-world application execution. We also introduce a generator-extractor baseline model for this task, which can adaptively select the source of the next token (i.e., from the vocabulary or from textual/visual contexts) for the executable command. This serves as a strong starting point while still leaving a large human-machine performance gap for useful future work.[1]

Figure 1: Conversational agent for image search and editing (CAISE). The dialogue starts with the image search request from the user. The assistant conducts the image search and addresses the following image search or editing requests for the user through 4 turns of request-execution exchange ([·] shows the image search/editing commands to the system).

1 Introduction

As the technology of image editing is developing and being refined, its utility is also increasing. It has become a usual practice to add editing effects to photos to make them look better. However, using image editing tools requires the expertise and skill that regular layperson users do not have. The names of these photo effects are not familiar and even the implication of the effects on images are not intuitive for most users. Hence, to increase the accessibility to these tools, proper individual expert guidance is required. However, guidance assistant systems run by small groups of available human experts could not cover all the requests from a large number of users worldwide. Instead, editing tools can benefit from having an automated assistant system that can have a conversation with users at scale to help them with their editing needs.

On the other hand, as the purpose and use cases of image editing are getting diverse, source materials for image editing also need to be diversified. For example, users might want to recreate their photos by adding additional objects from external sources. Users may also want to follow a reference image to make their photos more attractive (e.g., by borrowing a color from the source image). Hence, these activities call for an image search interface to be integrated

[1]Our code and dataset are publicly available at: https://github.com/hyounghk/CAISE

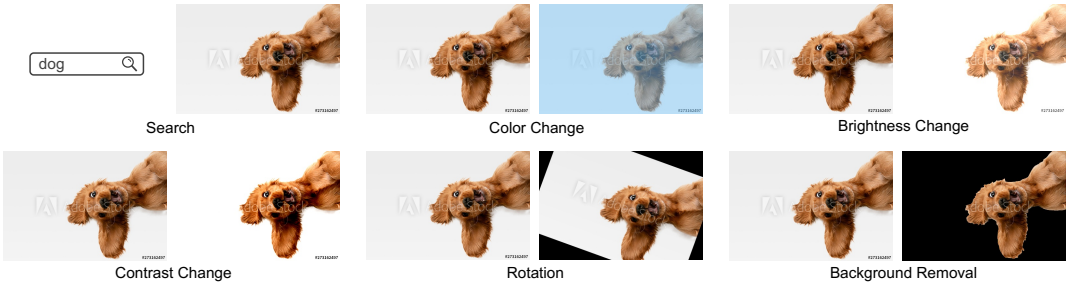

Figure 2: The diverse image search and editing effect functions that our CAISE dataset employs.

with image editing tools to provide a more integrated and comprehensive platform.

There have been some prior efforts towards automated image editing systems. They have focused on intent/action/goal identification from image editing requests (Manuvinakurike et al. 2018a,b,c; Lin et al. 2018), exploring low-level editing terms (Lin et al. 2020), editing images from descriptions (Shi et al. 2020), and describing image differences caused by image editing (Tan et al. 2019). However, there has been limited effort to integrate conversational image search and editing functions in a directly executable end-to-end manner for deployment into real-world applications. Therefore, we propose a new dataset, CAISE ('**C**onversational **A**gent for **I**mage **S**earch and **E**diting'), in which a user and an assistant hold a conversation in natural language (English) about image search and editing (Figure 1). The user's role is to make requests for image search and editing and the assistant's role is to search or edits images according to the user's requests and return the results while responding with natural language.

To collect such data, we implement a dialogue interface and ask pairs of annotators (one operating as the user and the other one as the assistant) to converse and search/edit images via the interface. The user is provided with multiple seed images from which they can get some ideas about what to search in the first place. Also, we show the user a list of suggested image search/editing functions to keep the command types diverse by asking them to follow the list as long as they can. The assistant annotator on the other hand, is equipped with an image search and editing interface to perform the user's requests. All command executions lead to the corresponding executable commands to be recorded. A total of 1.6K dialogues and 6.2K task instances are collected. The collected dialogues contain different types of image search/editing requests from users (direct request, implied request, object referring request; Table 3) and assistants' diverse responses (Sec.5), requiring models to understand the diverse grounded interactions in the conversations.

The task on the CAISE dataset is to generate the executable commands (e.g., search, color-change, brightness-change, contrast-change, rotation, background-removal; Figure 2) given the conversation and image contexts. This task setup simulates real-world image editing tools, facilitating important initial steps towards deployment in downstream applications. We introduce a novel generator-extractor model as a strong starting point baseline for this task and dataset. We employ a copying mechanism (Vinyals, Fortunato, and Jaitly 2015; Gu et al. 2016; Miao and Blunsom 2016; See, Liu, and Manning 2017), with which the model adaptively selects a way (i.e., generate from the vocabulary or extract from the context) to decode the next word since the clues for arguments of an executable command could be implicitly mentioned in the user's request (e.g., *"Please change the image color with **color of bus**"*) or the request contains the direct cues (e.g., *"Is it possible to increase the brightness of the image by **30** percent"*). For more effective model performance, we extend this mechanism so that it can also cover visual concepts in images by extracting object attributes or names from a set of object detection based concepts. For example, for the request *"Please change the image color with the color of bus"*, the corresponding color can not only be generated from the vocabulary, but also directly copied from one of object detection results, *"red bus"*. Our experiments show our baseline model performs effectively as a starting point, and we also demonstrate a large human-machine performance gap to allow useful future works on this important and understudied task.

Our contributions are two-fold: (1) we introduce a novel grounded dialogue dataset, CAISE, which incorporates image search and editing, featuring executable commands, hence allowing for more practical use in real-world applications. (2) We also introduce a generator-extractor model as a strong starting point baseline which extends the copy mechanism to the visual concept extraction, allowing for more effective performance and helping the interpretation of the model's behavior, while also leaving a large human-machine performance gap to allow useful future work by the community on this new challenging multimodal task.

2 Related Work

Image Editing. There have been some prior efforts to automate image editing programs. The research on image editing has been focused on intent identification (Manuvinakurike et al. 2018c), request to actionable command mapping (Manuvinakurike et al. 2018b; Lin et al. 2018), dialogue act labeling (Manuvinakurike et al. 2018a), low-level image edit requests (Lin et al. 2020), description to editing (Shi et al. 2020), or editing to description (Tan et al.

2019). Also, language-based image editing (Shinagawa et al. 2017; Chen et al. 2018; El-Nouby et al. 2019; Fu et al. 2020) focuses on an image generation task setup. However, there have been relatively few studies that pursue end-to-end conversational image editing agent systems combined with image search functionality. Our CAISE dataset supports the direct deployment of conversational image editing assistant systems by incorporating executable commands in the dataset, and also integrates image search functionality so as to make it more comprehensively useful.

Referring Expression Comprehension. Referring to an object using neighboring objects and relations between them is important to specify the object exactly and reduce ambiguities. Agents should have the ability to understand these expressions for better communication with humans or other agents. Referring expression comprehension has been studied actively (Kazemzadeh et al. 2014; Mao et al. 2016; Hu et al. 2016; Yu et al. 2018; Chen et al. 2019; Qi et al. 2020). Object referring expression plays an important role in image search and editing activities too. Users might need to specify an object or region that photo effects should be applied to, or want to search an item, of which they don't know the exact name, by referring it using spatial relations with other objects in an image. Our CAISE dataset contains a large amount of referring expressions to encourage agents to have the ability to understand such expressions.

Multimodal Dialogue. Multimodal dialogue has been actively studied in previous works (Das et al. 2017; De Vries et al. 2017; Mostafazadeh et al. 2017; Saha, Khapra, and Sankaranarayanan 2017; Pasunuru and Bansal 2018; Alamri et al. 2019; Haber et al. 2019; Kim et al. 2019; Moon et al. 2020; Shuster et al. 2020; Cheng et al. 2020). Although all these works involve interesting task setups (question answering/generation, object discovery, shopping, collaborative drawing, response retrieval/generation, image identification/generation, etc.) with different multimodal features (text, image, video, audio), there has not been focus on how to generate directly executable commands from the grounded multimodal dialogue. Moreover, to the best of our knowledge, our CAISE dataset and task is the first large-scale multimodal dialogue setup which combines the image search and editing tasks.

3 Task Description

Multimodal dialogue based executable command generation is one task that can be introduced from our CAISE dataset. Specifically, given a conversation history, previously searched and edited images, and previously executed commands, the agent should generate an executable command which can return the correct result for the user's request. The definitions of the executable commands are as follows:

Search. The search command retrieves images that are searched online with a query string. The format of the search command is [*search argument_1 ... argument_n ...*]. *'argument_n'* is the n-th token in the query string and there is no limit for the number of arguments.

Color Change. The color change command paints the whole image or a region of it with a designated color. The format of the color change command is [*adjust_color argument_1 argument_2*]. *'argument_1'* is a name of the colors (red, orange, green, blue, sky blue, purple, brown, yellow, pink), and *'argument_2'* is the value of intensity (0.0-1.0).

Brightness Change. The brightness change command changes the brightness of the whole image or a region of it with a designated intensity. The format of the brightness change command is [*adjust_attr brightness argument_1*]. *'argument_1'* is the value of intensity (-100-100%).

Contrast Change. The contrast change command changes the contrast of the whole image or a region of it with a designated intensity. The format of the contrast change command is [*adjust_attr contrast argument_1*]. *'argument_1'* is the value of intensity (0-100%).

Rotation. The rotation command rotates the whole image by a designated degree. The format of the rotation command is [*rotate argument_1*]. *'argument_1'* is the value of degree (0-360).

Background Removal. The background removal command makes the whole image black except the main subject. The format of the background removal command is [*image_cutout*]. There is no argument.

For illustrations of these photo effects, see Figure 2.

	Count	
	Per Dialogue	Total
Dialogue	-	1,611
Utterance	15.5	24,938
Utterance (user)	7.9	12,641
Utterance (assistant)	7.6	12,297
Executable Command	3.8	6,173
Image	3.8	6,173

Table 1: The number of dialogue components. Dialogues in our CAISE dataset are long (15.5 utterances) with four turns of image search/editing request-execution exchanges.

4 Dataset

Our CAISE dataset consists of conversations between a 'user' and an 'assistant'. Each conversation includes utterances of the user and assistant, searched or edited images, and executed commands.

Conversation Interface. We implement a dialogue system through which a pair of people chat about image search and editing. We build the user-side and the assistant-side interfaces separately since their roles are quite different. In the user-side interface, we provide 15 random seed images from COCO dataset (Lin et al. 2014) to help the user decide what to request for the first image search. We also present a suggestion for types of search and editing, which is a list of four commands from different types being randomly selected and ordered to avoid repeating the same search/editing order so that the user can follow it when they request to the assistant. In the assistant-side interface, we prepare a customized light-weight search and editing tool for the assistant to ad-

	Length				
	avg	stddev	median	max	min
Utterance	5.26	4.98	4.0	38	1
Utterance (user)	6.99	6.16	6.0	38	1
Utterance (assistant)	3.49	2.24	3.0	24	1

Table 2: The lengths of utterances in the dialogue collection. The user utterances are longer than assistant's due to the difference of their roles. The standard deviation of the lengths are large, indicating the utterances have various lengths.

dress the users' requests. We use Adobe Stock[2] for the image search engine, Adobe Photoshop[3] for the background removal function, and OpenCV[4] to implement the other editing functions. All the search and editing effects conducted from the tool are recorded in the form of executable commands that are used for corresponding functions. See supplementary material for the images of the interfaces.

Data Collection. We employ 10 annotators and train them to make them familiar with the collection interface and their primary roles, and guarantee the quality of the dataset. In the training session, we check all the practice dialogues manually and give feedback. We perform this training session multiple times until the quality of the dialogues gets above some threshold (see supplementary material for the detailed training process). After the training period, two annotators are paired so that one of them takes the user role and the other takes the assistant role. User-annotators are asked to give four requests throughout a conversation. Assistant-annotators are asked to perform the image search and editing functions according to the user-annotators' requests. If the user-annotators' requests are not clear, the assistant-annotators can ask them to clarify. We hire freelancers since the collection process needs some training to build expertise (especially for manipulating the search/editing interface), and pairing between the user and assistant annotators via a general crowd-sourcing platform is not easy.[5]

Payment. We pay up to 2 USD per dialogue, including bonuses. We also pay for dialogues which are created by annotators in their training period. Considering the time taken for a dialogue (around 5 minutes for a pair of trained annotators), the hourly wage is competitive (nearly 12 USD/hour per annotator).

5 Data Analysis

We collect 1,611 dialogues and create 6,173 task instances from the dialogue collection (since each dialogue has around four executable commands).

[2]https://www.adobe.io/apis/creativecloud/stock.html (the watermarks on the images are from using the Adobe Stock API).

[3]https://adobedocs.github.io/photoshop-api-docs-pre-release/

[4]https://opencv.org/

[5]We use Upwork (https://www.upwork.com) to hire freelancer annotators for high-quality, trained-expert human feedback. Upwork provides various communication tools (text chat and video/audio call interfaces) to facilitate communication with annotators and thus enable effective and efficient annotator training, as also shown in (Stiennon et al. 2020).

Type	Examples
Dir-Req	*"I was looking for an image of zoo"* *"Now increase the brightness of the image by 40 percent"* *"Please get rid of the background"*
Impl-Req	*"Can we repeat further by 130 degree more"* *"Can we try increasing further by 50 more"*
ObjRef-Req	*"Please find an image of the object seen to the right of the juicer in the above image"* *"Please change the color of image which matches with the color of cushion"*

Table 3: The examples of different types of requests (Dir-Req: direct request, Impl-Req: implied request, ObjRef-Req: object referring request).

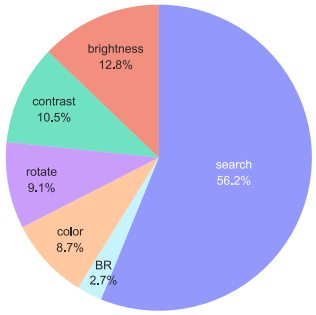

Figure 3: The executable commands frequency. The search command has the highest frequency since each dialogue begins with a search request (BR: background removal).

Dialogue Length. As shown in Table 1, the average number of utterances from both the users and assistants is 15.5 (7.9 and 7.6 from users and assistants, respectively). The average number of executable commands and images are the same (3.8 per dialogue) since each image is the result of the execution of each corresponding command.

Utterance Length. As shown in Table 2, the average length of user utterances is larger than that of assistant utterances (6.99 vs. 3.49). The reason is that user utterances are mainly about image search and editing requests, requiring detailed explanations (e.g., *"Could you also find me an image of dress for my wife matching the color of hat in the above image?"*). On the other hand, assistant utterances are usually short responses to users' requests (e.g., *"okay"*, *"sure"*) or questions for users' confirmation (e.g., *"Do you like it?"*, *"Is this fine?"*), and clarifications (e.g., *"clock wise or anti clockwise?"*). Also, the standard deviations of the utterance lengths are large compared to the average lengths, confirming utterances in our CAISE dataset have various lengths.

User Request Types. As shown in Table 3, we can categorize the image search and editing requests mainly into three types: direct request, implied request, and object referring request. Direct requests are self-contained requests which have direct clues about what users are asking. Implied requests are the ones that do not explicitly mention what types of functions are asked to be performed but imply them from

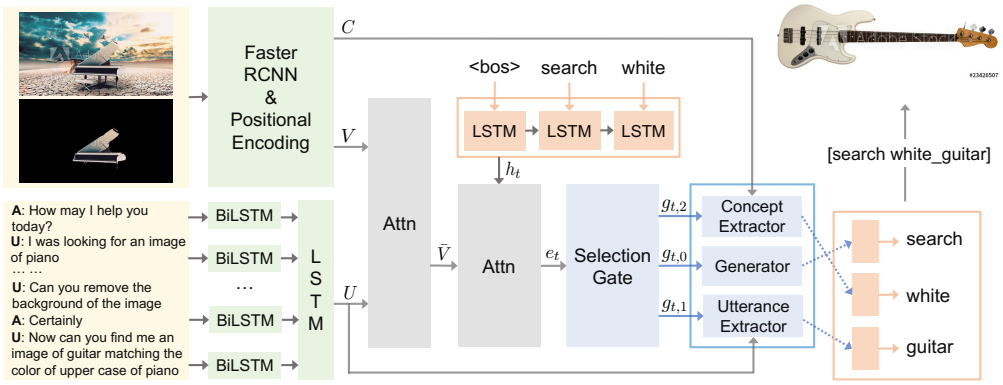

Figure 4: The Generator-Extractor model. The model adaptively selects the source of the next token via the selection gate (for the simplicity purpose, some blocks and relationship arrows are omitted; input "<bos> search white" to the LSTM block is previously generated tokens, i.e., an autoregressive decoding setup; the model produces the command word-by-word).

the conversation contexts. Object referring requests are the ones that use the information of objects (i.e., color, name, location) in images to specify what should be done.

Assistant Response Types. Although several assistant responses are generic confirmation-based (since the assistants' main role is to perform image search and editing according to users' requests), there are also several other types of interesting responses such as correction (user: *"... image of sea-saw ..."* - assistant: *"Do you mean see-saw?"*), ambiguity-clarification (user: *"... rotate the image to 40 degree"* - assistant: *"... clock wise or anti clockwise?"*), coreference (assistant: *"How would you like it by"*), etc., encouraging models to understand the diverse grounded interactions in the conversations between users and assistants to perform the task.

Executable Commands Frequency. As shown in Figure 3, the search command is the most frequent. The reason is that search is the first command that must be performed in every dialogue, and we design the collection interface so that each dialogue has one additional search command on average (the ratio of the first-line search commands to the other search commands is 46.5% vs. 53.5%). The low frequency of the background removal command ("BR" in the figure) is due to the command's instability. Unlike the other commands that do not fail, the background removal command could fail depending on images' contents (it seems that images that have complicated contents are hard to remove background from). Once the background removal command fails, the user-annotators might not try it again and perform one of the other functions instead.

6 Models

We present the generator-extractor model as a starting point baseline (Figure 4). The model takes the history of utterances, images, and previously executed commands as input, and predicts a next executable command.

Encoder. A large part of our CAISE dataset involves objects and their concepts (names and attributes) in images, especially for the search command. Thus, we employ Faster R-CNN (Girshick 2015) to extract object visual features $\hat{V}$, bounding box features B, and their concept features W^c, which are usually made of a couple of tokens, from images I. $\hat{V}$ and B are combined through a linear layer, and W^c is further encoded by a word embedding layer and the bidirectional LSTM (Hochreiter and Schmidhuber 1997):

$$\hat{V}, B, W^c = \text{FRCNN}(I), \quad V = \text{PE}(\text{Linear}([\hat{V}; B])) \quad (1)$$

$$\hat{C} = \text{Embed}(W^c), \quad C = \text{PE}(\text{BiLSTM}(\hat{C})) \quad (2)$$

where PE denotes positional encoding (Gehring et al. 2017; Vaswani et al. 2017) and it is applied image-wise (i.e., the same encoding value is applied to the features from the same image). Sequences of tokens from utterances W^u in dialogue D are encoded by the bidirectional LSTM, and the last forward hidden state and the first backward hidden state of $\hat{U} \in \mathbb{R}^{M \times N \times d}$ are extracted and concatenated to create a vector which represents each utterance, where M is the dialogue length, N is the utterance length, and d is the feature dimension. Then, the sequence of the utterance features is fed to a LSTM to learn the dialogue context:

$$\hat{U} = \text{BiLSTM}(\text{Embed}(W^u)) \quad (3)$$

$$U = \text{LSTM}([\hat{U}^f_{N-1}; \hat{U}^b_0]) \quad (4)$$

We employ the attention mechanism to align the visual features V, and utterance features $U \in \mathbb{R}^{M \times d}$. We calculate the similarity matrix $S \in \mathbb{R}^{O \times M}$ between visual and utterance features, where O is the total number of all object features from the images: $S_{ij} = V_i^\top U_j$. From the similarity matrix, the new fused visual and utterance feature is:

$$\bar{U} = \text{softmax}(S) \cdot U, \quad \bar{V} = [V; \bar{U}; V \odot \bar{U}] \cdot W_v \quad (5)$$

where $W_v \in \mathbb{R}^{3d \times d}$ is the trainable parameter, $\odot$ is element-wise product, and $\cdot$ is matrix multiplication. Tokens from an executable command, $\{w_t\}_{t=1}^T$, are embedded in the embedding layer, and then sequentially fed to the LSTM layer.

$$\hat{w}_{t-1} = \text{Embed}(w_{t-1}), \quad h_t = \text{LSTM}(\hat{w}_{t-1}, h_{t-1}) \quad (6)$$

The same (but with different parameters) attention mechanism (Attn), which is applied to visual and utterance features, is used for aligning the command feature, h_t, and $\bar{V}$.

$$e_t = \text{Attn}(h_t, \bar{V}) \quad (7)$$

	Models	Accuracy (%)						
		total	search	color	brightness	contrast	rotation	remove-back
1	Base	22.33	11.45	28.63	32.13	46.46	9.52	**100.0**
2	Base+VE	22.12	11.00	30.77	30.12	48.56	8.93	**100.0**
3	Base+UE	45.23	36.42	26.07	**49.80**	92.13	**29.17**	97.14
4	Base+UE+VE	**46.43**	**37.43**	**40.60**	48.39	**93.18**	26.49	97.14
5	Human Expert	90.0	82.0	90.0	100.0	100.0	100.0	100.0

Table 4: Model performance on the test split. The extractors help improve the model's performance (Base: the basic encoder-decoder model only with generator (without extractors), UE: utterance extractor, VE: visual concept extractor).

	Models	Accuracy (%)
1	Request-Only	42.30
2	DialogHistory-Only	0.66
3	Request+DialogHistory	43.17
4	Vision-Only	0.93
5	Request+Vision	45.56
6	Request+DialogHistory+Vision	46.43

Table 5: Modality ablations. Each modality/component helps improve the model's performance.

Generator. The generator calculates the probability of each token in the vocabulary that contains all possible candidates.

$$l_t = \text{Linear}(e_t), \quad a_t^g = \text{softmax}(l_t) \tag{8}$$

Extractor. Utterances in our CAISE dataset contain many direct clues for generating commands. Thus, the model would benefit from extracting keywords from the context. We employ a copying mechanism (Vinyals, Fortunato, and Jaitly 2015; Gu et al. 2016; Miao and Blunsom 2016; See, Liu, and Manning 2017) to implement the extraction.

$$(A_t^u)_i = h_t^\top U_i, \quad a_t^u = \text{softmax}(A_t^u) \tag{9}$$

The model also can directly obtain useful information from the visual concept since visual concept features can provide object names/attributes in a textual semi-symbolic format.

$$(A_t^c)_i = e_t^\top C_i, \quad a_t^c = \text{softmax}(A_t^c) \tag{10}$$

Selection Gate. To adaptively select the source of the next token, we employ gating approach (See, Liu, and Manning 2017) to obtain the adaptive weights:

$$g_t = \text{softmax}(W_g^\top e_t) \tag{11}$$

where $W_g \in \mathbb{R}^{d\times 3}$ is the trainable parameter. The weighted sum of each probability from each source is the final probability of the next token.

$$p(w_t|w_{1:t-1}, I, D) = g_{t,0} \cdot a_t^g + g_{t,1} \cdot a_t^u + g_{t,2} \cdot a_t^c \tag{12}$$

The loss is:

$$L = -\sum_{t=1}^{T} \log p(w_t^*|w_{0:t-1}, I, D) \tag{13}$$

where w_t^* is the GT token.

7 Experiments

Data Splits. We split the total 1,611 dialogues into 1,052, 262, and 297 for train, validation, and test set, respectively. From the dialogue splits, we obtain 4,059/1,002/1,112 (train/valid/test) instance splits.

Evaluation Metric. We use accuracy as the evaluation metric. For image search and editing systems, it is important to feed the correct command, and automatic metrics for text generation tasks such as BLEU (Papineni et al. 2002) are not appropriate. So, we only count generated commands which exactly match the ground-truth commands (i.e., command types and their arguments) as the correct ones. For the search command, generated commands with different query word orders (e.g., [*search juice glass*] and [*search glass juice*]) are also considered correct since queries with different word orders usually return the same or similar outcomes. For the color change command, we only compare the command type and color names but not up to intensity (e.g., [*adjust_color blue*]) since, in most cases, users ask to change colors without saying a specific value of intensity (e.g., *"Color the image to the same color as the salmon in the above image"*).

Human Expert Performance. We randomly sample 50 instances for the search command and 10 instances for each of the other commands (total 100 instances) and ask an expert who knows the task well to predict the commands based on the textual and visual context.

Training Details. We use 512 as the hidden size and 256 as the word embedding dimension. We use Adam (Kingma and Ba 2015) as the optimizer with the learning rate 1×10^{-4}. See supplementary material for more details.

8 Results

As shown in Table 4, the extractor modules help improve the model's performance. The utterance extractor helps much to improve the model's performance (row 1 and 3). Especially, the scores for the search, brightness change, contrast change, and rotation commands get increased, implying that the utterance extractor can effectively locate the direct clues from the dialogue history context. Applying the visual concept extractor additionally increases the score (rows 3 and 4).[6] The performance of the search and color change commands gets improved from this application, meaning that the visual concept extractor can match the visual features and the concept

[6]The stddev of the full model (Base+UE+VE) scores is 0.74, and the score of the model on validation split is 49.7%.

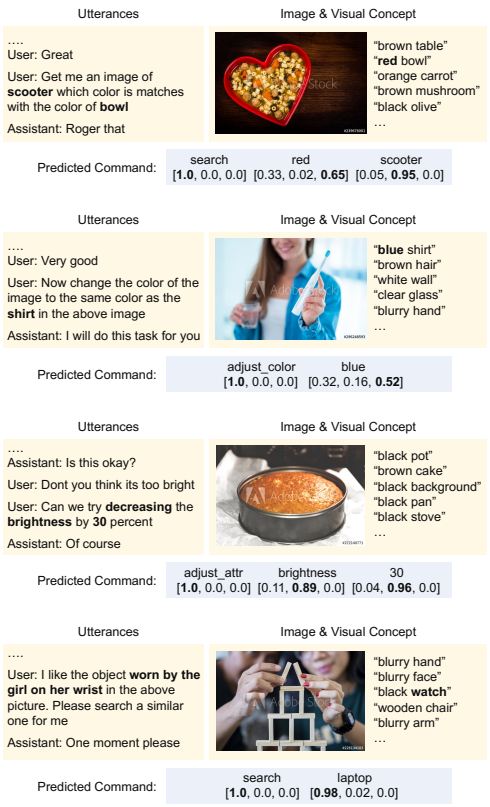

Figure 5: The examples of the model output (1st and 2nd examples: correct / 3rd and 4th: incorrect). Our model can effectively use the generator and extractors by selecting them with the adaptive selection gate (the numbers in bracket are the selection gate weight, i.e., [weight for the generator, weight for the utterance extractor, weight for the visual concept extractor]). The bottom two figures show incorrect examples in which the model cannot figure out the meaning of *'decreasing'* and cannot catch *'watch'* from the image.

features, and align them with requests. But, when comparing rows 1 and 2, adding the visual concept extractor to the base model does not seem to help. Although it shows a similar improvement pattern for the other commands, the performance for the search command is not improved. That implies that the visual concept extractor is effective together with the utterance extractor (see the example at the top of Figure 5).[7]

Human Expert Performance. As shown in row 4 and 5 of Table 4, the human-machine performance gaps are large for most of the command types, implying that there is large room for future work to develop novel improvements on this new multimodal dialogue task, and our baseline described above is meant to serve as a strong starting point.

Modality Ablation. Table 5 shows the ablation results from different combinations of the model (Base+UE+VE) components. We take the last two utterances from the dialogue as 'request' since there is no explicit division between request and context in our CAISE dataset. As shown in row 2 and 4, the model could not perform well without the 'request'. That is obvious since, without this information, the model cannot figure out what and how to search and edit. The request-only (row 1) records a high score possibly because many of the requests contain direct clues like *"Can you rotate the image counterclockwise by 30 degrees"*. Adding dialogue history (row 1 and 3, row 5, and 6) helps, meaning the request needs dialogue context for better performance. Also, adding visual context (images) improves the model's performance (row 1 and 5, 3 and 6) because there are requests (such as for the search and color change commands) that need to refer to objects/colors in the visual context to be performed correctly.[8]

Output Examples. Figure 5 shows examples of the model output. In the top figure, our model gives the correct command ([*search red scooter*]) according to the request. Specifically, the model generates the command name, *'search'*, using the generator (with the selection gate weight of 1.0), extracts the color, *'red'*, using the visual concept extractor (with the weight of 0.65), and also extracts the item name to search for, *'scooter'* using the utterance extractor (with the weight of 0.95). The second figure shows the example of the color change command. The model also generates the correct command name, *'adjust_color'* using the generator (with the weight of 1.0). The model then extracts the color, *'blue'*, from the visual concept feature using the visual concept extractor (with the weight of 0.52). On the other hand, as shown in the third figure, the model fails to understand the meaning of *'decreasing'* and just extracts *'30'* using the utterance extractor (with the weight of 0.96) for intensity (the ground-truth value is -30). In the bottom figure, the model cannot catch *'watch'* in the image and generated the wrong searching query, *'laptop'* using the generator (with the weight of 0.98). These negative results from our baseline model imply that there is room for improvement via more advanced modeling approaches in future work from the community on our CAISE task.

9 Conclusion

We introduced a novel conversational image search and editing task/dataset, called CAISE, in which an agent should conduct image search and editing according to users' requests. To implement and train the automated system, we collected a dialogue dataset in which a user and an assistant hold a conversation on image search/editing. We presented the generator-extractor model as a strong starting point baseline and the large human-machine performance gap showed there is room for improvement on this task.

[7] While we evaluate the performance via the average score over each search/editing instance like in (Das et al. 2017), one other possible evaluation option for practical applications is to consider the average success rate of the whole search/editing dialogue (5.4% from our full (Base+UE+VE) model).

[8] We randomly sample 75 instances (except the first-turn search command) and conduct human evaluation on which inputs are required to perform the requests. Request-only: 43%; need-DialogHistory+Vision: 57% (need-DialogHistory 13%, need-Vision 47%, need-both 3%). This means that to solve our command generation task, models need to understand the context (we observe the same trend when we also include the first search command).

Acknowledgments

We thank the reviewers for their helpful comments. This work was partially done while HK was interning at Adobe Research and later extended at UNC, where it was supported by NSF Award 1840131, ARO-YIP Award W911NF-18-1-0336, DARPA KAIROS Grant FA8750-19-2-1004, and a Google Focused Award. The views contained in this article are those of the authors and not of the funding agency. This work was done while DK was at Adobe Research.

References

Alamri, H.; Cartillier, V.; Das, A.; Wang, J.; Cherian, A.; Essa, I.; Batra, D.; Marks, T. K.; Hori, C.; Anderson, P.; et al. 2019. Audio visual scene-aware dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7558–7567.

Chen, H.; Suhr, A.; Misra, D.; Snavely, N.; and Artzi, Y. 2019. Touchdown: Natural Language Navigation and Spatial Reasoning in Visual Street Environments. In *Conference on Computer Vision and Pattern Recognition*.

Chen, J.; Shen, Y.; Gao, J.; Liu, J.; and Liu, X. 2018. Language-based image editing with recurrent attentive models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8721–8729.

Cheng, Y.; Gan, Z.; Li, Y.; Liu, J.; and Gao, J. 2020. Sequential attention GAN for interactive image editing. In *Proceedings of the 28th ACM International Conference on Multimedia*, 4383–4391.

Das, A.; Kottur, S.; Gupta, K.; Singh, A.; Yadav, D.; Moura, J. M.; Parikh, D.; and Batra, D. 2017. Visual Dialog. In *CVPR*.

De Vries, H.; Strub, F.; Chandar, S.; Pietquin, O.; Larochelle, H.; and Courville, A. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5503–5512.

El-Nouby, A.; Sharma, S.; Schulz, H.; Hjelm, D.; Asri, L. E.; Kahou, S. E.; Bengio, Y.; and Taylor, G. W. 2019. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In *Proceedings of the IEEE International Conference on Computer Vision*, 10304–10312.

Fu, T.-J.; Wang, X. E.; Grafton, S.; Eckstein, M.; and Wang, W. Y. 2020. SSCR: Iterative Language-Based Image Editing via Self-Supervised Counterfactual Reasoning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. In *ICML*, 1243–1252.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1631–1640.

Haber, J.; Baumgärtner, T.; Takmaz, E.; Gelderloos, L.; Bruni, E.; and Fernández, R. 2019. The PhotoBook Dataset: Building Common Ground through Visually-Grounded Dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1895–1910.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Hu, R.; Xu, H.; Rohrbach, M.; Feng, J.; Saenko, K.; and Darrell, T. 2016. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4555–4564.

Kazemzadeh, S.; Ordonez, V.; Matten, M.; and Berg, T. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 787–798.

Kim, J.-H.; Kitaev, N.; Chen, X.; Rohrbach, M.; Zhang, B.-T.; Tian, Y.; Batra, D.; and Parikh, D. 2019. CoDraw: Collaborative Drawing as a Testbed for Grounded Goal-driven Communication. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6495–6513.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Lin, T.-H.; Bui, T.; Kim, D. S.; and Oh, J. 2018. A multimodal dialogue system for conversational image editing. *Workshop in NeurIPS*.

Lin, T.-H.; Rudnicky, A.; Bui, T.; Kim, D. S.; and Oh, J. 2020. Adjusting Image Attributes of Localized Regions with Low-level Dialogue. In *Proceedings of The 12th Language Resources and Evaluation Conference*, 405–412.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Manuvinakurike, R.; Brixey, J.; Bui, T.; Chang, W.; Artstein, R.; and Georgila, K. 2018a. Dialedit: Annotations for spoken conversational image editing. In *Proceedings 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, 1–9.

Manuvinakurike, R.; Brixey, J.; Bui, T.; Chang, W.; Kim, D. S.; Artstein, R.; and Georgila, K. 2018b. Edit me: A corpus and a framework for understanding natural language image editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Manuvinakurike, R.; Bui, T.; Chang, W.; and Georgila, K. 2018c. Conversational image editing: Incremental intent identification in a new dialogue task. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 284–295.

Mao, J.; Huang, J.; Toshev, A.; Camburu, O.; Yuille, A. L.; and Murphy, K. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 11–20.

Miao, Y.; and Blunsom, P. 2016. Language as a Latent Variable: Discrete Generative Models for Sentence Compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 319–328.

Moon, S.; Kottur, S.; Crook, P. A.; De, A.; Poddar, S.; Levin, T.; Whitney, D.; Difranco, D.; Beirami, A.; Cho, E.; et al. 2020. Situated and Interactive Multimodal Conversations. *COLING*.

Mostafazadeh, N.; Brockett, C.; Dolan, B.; Galley, M.; Gao, J.; Spithourakis, G.; and Vanderwende, L. 2017. Image-Grounded Conversations: Multimodal Context for Natural Question and Response Generation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 462–472.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 311–318.

Pasunuru, R.; and Bansal, M. 2018. Game-Based Video-Context Dialogue. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 125–136.

Qi, Y.; Wu, Q.; Anderson, P.; Wang, X.; Wang, W. Y.; Shen, C.; and van den Hengel, A. 2020. REVERIE: Remote Embodied Visual Referring Expression in Real Indoor Environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Saha, A.; Khapra, M.; and Sankaranarayanan, K. 2017. Towards building large scale multimodal domain-aware conversation systems. *AAAI*.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1073–1083.

Shi, J.; Xu, N.; Bui, T.; Dernoncourt, F.; Wen, Z.; and Xu, C. 2020. A Benchmark and Baseline for Language-Driven Image Editing. *Asian Conference on Computer Vision (ACCV)*.

Shinagawa, S.; Yoshino, K.; Sakti, S.; Suzuki, Y.; and Nakamura, S. 2017. Interactive image manipulation with natural language instruction commands. *NeurIPS Workshop*.

Shuster, K.; Humeau, S.; Bordes, A.; and Weston, J. 2020. Image-chat: Engaging grounded conversations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2414–2429.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33.

Tan, H.; Dernoncourt, F.; Lin, Z.; Bui, T.; and Bansal, M. 2019. Expressing Visual Relationships via Language. In *ACL*, 1873–1883.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *Advances in neural information processing systems*, 2692–2700.

Yu, L.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; Bansal, M.; and Berg, T. L. 2018. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1307–1315.