Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 405–412
Marseille, 11–16 May 2020

Adjusting Image Attributes of Localized Regions with Low-level Dialogue

Tzu-Hsiang Lin[1], Alexander Rudnicky[1], Trung Bui[2], Doo Soon Kim[2], Jean Oh[1]
[1]Carnegie Mellon University, [2]Adobe Research
tzuhsial@alumni.cmu.edu, {air,jeanoh}@cmu.edu, {bui,dkim}@adobe.com

Abstract

Natural Language Image Editing (NLIE) aims to use natural language instructions to edit images. Since novices are inexperienced with image editing techniques, their instructions are often ambiguous and contain high-level abstractions which require complex editing steps. Motivated by this inexperience aspect, we aim to smooth the learning curve by teaching the novices to edit images using low-level command terminologies. Towards this end, we develop a task-oriented dialogue system to investigate low-level instructions for NLIE. Our system grounds language on the level of edit operations, and suggests options for users to choose from. Though compelled to express in low-level terms, user evaluation shows that 25% of users found our system easy-to-use, resonating with our motivation. Analysis shows that users generally adapt to utilizing the proposed low-level language interface. We also identified object segmentation as the key factor to user satisfaction. Our work demonstrates advantages of low-level, direct language-action mapping approach that can be applied to other problem domains beyond image editing such as audio editing or industrial design.

Keywords: dialogue, image editing, natural language understanding, low-level

1. Introduction

Image editing has long been in demand since the invention of photography. However, learning image editing is time-consuming. It involves a wide assortment of features, and combinations of these features to achieve a desired effect. As of today, most novices edit their images by requesting experts for help. These requests come in the form of Image Edit Requests (IER) – natural language descriptions which express desired changes to be made. When experts see these requests, they can accurately edit these images via a two-stage process: (i) First, experts look at the original image and the IER and interprets the high-level concepts expressed by the novice. (ii) Second, they come up with one or more low-level edit operations for these concepts and apply these edits. Can we build machines that do the same thing? This motivates the study on Natural Language Image Editing (NLIE).

There are mainly two approaches towards NLIE. The two-stage approach follows the editing process of experts. Several works have developed semantic parsers for the first stage, with datasets collected from crowdsourcing (Manuvinakurike et al., 2018a), online image editing communities (Mohapatra, 2018), or spoken conversation with experts (Manuvinakurike et al., 2018b). Though these semantic parsers are able to capture the high-level intent of novices, little has been discussed about the second stage – how to infer edits with these parsers. These datasets still require expert annotations for generating edits, which is a difficult one-to-many mapping. On the other hand, the image generation approach directly generates an edited image given the original image and an IER using the end-to-end adversarial learning (Goodfellow et al., 2014) framework. This approach is appealing as it learns image edits end-to-end, but also suffers from major drawbacks which limits the number of edit features it can support. (i) First, image generation is data hungry and domain specific. To support a new edit feature, a new dataset needs to be collected. For example, a model trained to modify color of flowers cannot modify color of birds (Chen et al., 2018). (ii) Second, image generation is mostly restricted to global manipulations (Wang et al., 2018). Editing a particular region loses the end-to-end appeal as it requires a separate segmentation module (Lin et al., 2014), bringing it back to the two-stage approach.

One major challenge we observed in NLIE is the novice language. Having little or no knowledge of terminologies and techniques, novices use open-domain vocabulary to express their needs. As a result, novice IERs have certain characteristics: (i) ambiguous, (ii) abstract, and (iii) imprecise. Though experts are able to disambiguate or fill-in missing details for novice IERs, grounding open-domain vocabulary have posed difficulties for NLIE. For the two-stage approach, annotators have low or near chance level agreement on certain entities (Brixey et al., 2018), and datasets collected under different scenarios require different annotation schemas (Manuvirakurike et al., 2018). For the image-generation approach, IERs are often imprecise and includes multiple edit operations in a single request (e.g., enhance white balance and contrast) (Wang et al., 2018). If novices have some knowledge of image editing tools, machines could understand IERs more easily and will be able to perform edits more precisely.

In this paper we investigate the potential of low-level language for image editing. Motivated by the difficulty of open-domain vocabulary, we propose to use dialogue to bridge novice language (open-domain vocabulary) to image editing terminologies (in-domain vocabulary). Our hypothesis is that, dialogue interaction is an accessible way for users to learn these in-domain low-level terms. Our end goal is different from previous works as they aim to develop conversational agents (Manuvinakurike et al., 2018b; Cheng et al., 2018) to complete image edits for users, while we focus on smoothing the learning curve for novices.

To validate our hypothesis, we developed an image editing dialogue system that grounds on terminologies and conducted a study of 83 users. Our system is able to segment common objects using referring expressions (Dale and Reiter, 1995), and adjust image attributes of these segmented regions. Users interact with our system via typed text, and

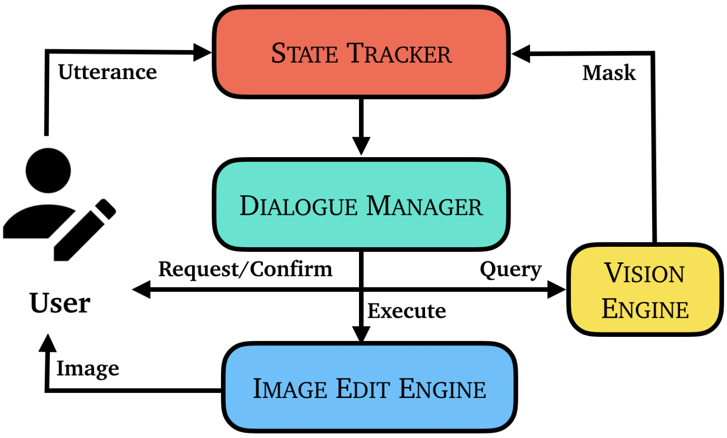


Figure 1: Illustration of our system architecture. Our system contains 4 components: (i) STATE TRACKER, (ii) DIALOGUE MANAGER, (iii) VISION ENGINE, (iv) IMAGE EDIT ENGINE. Users interact with our system using text inputs.

can see the immediate effects of their expressed edits. Our system received a *Fair* to *Good* rating, and users generally liked the smooth experience of dialogue. Through our analysis and user's feedback, we identified object segmentation as the key factor that leads to user (dis-) satisfaction. We further proposed *Vision Accuracy*, a metric based on dialogue act which correlates with user ratings.
Overall, our contributions can be summarized as follows:

- We show that combining dialogue with low-level IERs is a promising solution for supporting NLIE.
- Our analysis and collected dialogues provide insights on user behavior and system aspects of image editing. Additionally, we propose a novel metric that correlates with user ratings and discuss the strengths and limitations of our system.
- We collected two novel datasets to promote future research: (i) low-level IERs (ii) low-level image editing dialogues. Also, our modular system can be easily extended to support more image edits and be used to collect image editing dialogues.

2. Related Work

2.1. Dialogue

It is easy to imagine a dialogue for information-seeking domains (Raux et al., 2005; Bohus and Rudnicky, 2009; Dhingra et al., 2017; Wen et al., 2017; El Asri et al., 2017; Budzianowski et al., 2018) such as restaurant booking or movie booking, as these conversations exist in everyday life. Nevertheless, the same could not be said for image editing, as users interact with software tools (without language input) (Bychkovsky et al., 2011) or post on online communities (without multi-turn interaction) (Mohapatra, 2018; Tan et al., 2019). For NLIE, several works have crowdsourced novice IERs by giving image or paired images (Manuvinakurike et al., 2018a; Wang et al., 2018; Cheng et al., 2018). However, they did not consider image editing tools while collecting, and their IERs are often free form in expression. The most similar data to our work is the human-to-human dialogues collected in Conversational Image Editing (Manuvinakurike et al., 2018b), where novice uses only spoken language and an expert carries out the edits. Our work explores the potential of low-level language for image editing dialogues, focusing on smoothing the learning curve.

2.2. Language and Vision

There are many research fields which learns to align semantics between visual information and language, usually with an image and a natural language description. Our work falls under the category of a more recent setup, where language is related to the difference between paired images. Several works have been proposed for edited image generation using attention for alignment (Chen et al., 2018; Wang et al., 2018; Cheng et al., 2018). Other works focus more on the language side, either by determining whether a caption is true considering a pair of images (Suhr et al., 2019), or generating captions that describes the difference with latent variables (Jhamtani and Berg-Kirkpatrick, 2018) or attention (Tan et al., 2019). Our system is simpler as it grounds language at a low-level, and processes visual information with an independent segmentation module.

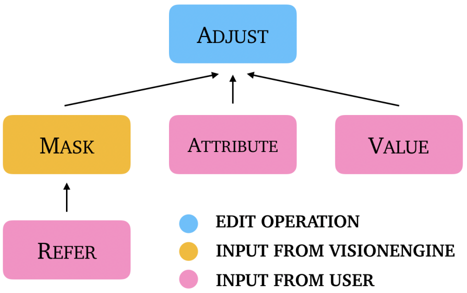


Figure 2: Domain ontology of our system. ADJUST operation depends on slots MASK, ATTRIBUTE, and VALUE. Slot MASK further depends on slot REFER.

3. Task

3.1. Definition

We define the task as instructing the system to perform image edit operations using natural language. To perform this task, the system needs to extract slot values from user's natural language inputs, locate the specified region in the image, and execute the requested edit.

3.2. Domain Ontology

ADJUST, which involves manipulating the value of some image property, is the most frequent edit in image editing and is usually realized with a slider; some editing tools provide only ADJUST operations (Bychkovsky et al., 2011). In our system, we study only the ADJUST edit, as it already includes image properties which novices typically have no knowledge about.

To perform ADJUST, three slot arguments are required (Figure 2). The first slot is MASK, which is a particular region in an image the user wants to modify, and further depends on slot REFER. This is a referring expression (Krahmer and Van Deemter, 2012) given by the user for the system to identify an object in an image. The second slot is ATTRIBUTE, which is a modifiable property of an image. The third is VALUE, which is the degree of manipulation of an ATTRIBUTE. We include 5 common ATTRIBUTES: $\{$*brightness, contrast, hue, saturation, lightness*$\}$ (Foley et al., 1996). The range of VALUE is from -100 to 100.

4. Natural Language Understanding

We first describe the key component of our system, which is Natural Language Understanding (NLU). To build our NLU, we collected low-level IERs (Section 4.1.) and trained a BIO tagger (Section 4.2.) to parse REFER, ATTRIBUTE, VALUE.

4.1. ILLC-IER Dataset

Image Edit Request (IER) has a broad definition and comes in many different forms. To ensure that the NLU corpus contains instances of all slots in the ontology, we collected sentences that instructs an image edit in imperative form, and includes all the low-level edit arguments. We refer to these specific type of sentences as "Imperative Low-Level Complete Image Edit Requests (ILLC-IERs)".

Collection We crowd-sourced ILLC-IERs using the Amazon Mechanical Turk (AMT) platform. We asked workers to provide a sentence that specifies an image edit and contains REFER, ATTRIBUTE, VALUE elements. For these slots, we provide (i) an image with a highlighted object, (ii) several referring expressions of the highlighted object, (iii) a randomly sampled ATTRIBUTE, and (iv) a randomly sampled VALUE. We used the RefCOCO dataset (Kazemzadeh et al., 2014) for (i) and (ii), and sampled 900 from train, 100 from dev, and 100 from test.

Annotation Similar to the data collection, we also crowd-sourced annotations using AMT. We choose 4 categories for ILLC-IER BIO tagging (i) ACTION (ii) REFER (iii) ATTRIBUTE (iv) VALUE. REFER, ATTRIBUTE, VALUE are slots in our ontology. ACTION is the imperative verb that corresponds to the edit action (e.g., "increase", "decrease", "modify") and indicates the sign of VALUE. For example, "decrease brightness by 10" is equivalent to "change brightness by -10".

Statistics We collected 2,537 ILLC-IERs. Based on the data splits in RefCOCO, we split them into 2,055 for train, 242 for dev, and 240 for test. The average number of words per IER is 12.7, and the average length of the referring expressions is 5.4. The total number of tokens is 32,194, and the number of unique tokens is 1,034.

4.2. Training BIO Tagger

We preprocessed ATTRIBUTE and VALUE by replacing them with abstract tokens `<attribute>` and `<value>` since these 2 slots have a closed set of values. We then trained a one-layer LSTM (Hochreiter and Schmidhuber, 1997) BIO tagger that reached a mean 98.69 F1 score on the test set. We also compiled a list of ACTION words that indicate a negative VALUE (e.g., "decrease", "lower", "reduce").

Note that ILLC-IER includes all slot values in a single sentence, which imposes a high degree of constraint, and largely limits the number of ways that users can express IERs. This establishes our grounding of in-domain vocabulary.

5. System

Our system architecture is based on the multimodal system of (Lin et al., 2018) and consists of 4 components: (i) STATE TRACKER, (ii) DIALOGUE MANAGER, (iii) VISION ENGINE, (iv) IMAGE EDIT ENGINE, along with a web-based interface. The main differences between the work of (Lin et al., 2018)'s and ours are: (i) their focus is on policy learning with multimodal dialogue state, while we investigate whether users would like the experience of low-level image editing dialogue. (ii) On the technical side, their system is multimodal which also allows gesture input, while ours only allows text inputs. However, their NLU is based on regular expressions, while ours is based on recurrent neural networks. Figure 1 shows the system diagram with dialogue actions. We describe each component in detail below.

5.1. State Tracker

STATE TRACKER contains the (i) NLU (Section 4.) component that extracts slot values, and a (ii) STATE UPDATER that aggregates NLU outputs over multiple turns. At every turn, STATE TRACKER takes the current user utterance and the previous turn's dialogue state, and outputs the dialogue state for the current turn. This dialogue state is then passed to DIALOGUE MANAGER.

NLU Our NLU is based on the LSTM tagger in Section 4., with the addition of a string matcher that detects $\{$*yes*, *no*$\}$ intents. The latter handles responses when DIALOGUE MANAGER (Section 5.2.) decides to *Confirm* and expects a *yes* or *no* input (e.g., "Is the current detected region correct? (yes/no)").

State Updater STATE UPDATER aggregates turn level outputs from NLU, and clears all slot values when DIALOGUE MANAGER decides to EXECUTE an adjustment. It

Dataset	ACTION	ATTRIBUTE	REFER	VALUE	MEAN
ILLC-IER	96.39	100.0	98.05	100.0	98.69
Dialogue	63.15	94.59	63.15	99.69	84.13

Table 1: NLU F1 scores on ILLC-IER dataset (Section 4.) and user study dialogues (Section 6.).

Rating	VERY POOR	POOR	FAIR	GOOD	EXCELLENT
Users	6	22	22	29	4

Table 2: Performance ratings by users. Our system received a *Fair* to *Good* rating.

also removes MASK value when a new REFER value is detected by NLU. For simplicity, we assume that all edits are independent, and a new REFER should have a new MASK.

5.2. Dialogue Manager

DIALOGUE MANAGER takes the current dialogue state (from STATE TRACKER) and outputs an action. The actions are: (i) *Request*. (ii) *Confirm* (iii) *Query* (iv) *Execute*. *Request* asks the user for a slot value. *Confirm* asks whether the current tracked slot value is correct. *Query* passes the current tracked REFER to VISION ENGINE and receives a MASK. *Execute* sends the tracked MASK, ATTRIBUTE, and VALUE to IMAGE EDIT ENGINE for execution.

Policy In image editing, users will commonly identify a region they want to modify, then decide on the types of edit. Based on this observation, we use a rule-based policy that obtains slots in the order of REFER, MASK, ATTRIBUTE, and VALUE. At the beginning of a dialogue, our DIALOGUE MANAGER will first *Request* the slot REFER. After obtaining REFER, it will *Query* the VISION ENGINE then *Confirm* whether the returned MASK is correct. Finally, it will *Request* the slots ATTRIBUTE and VALUE from the user and then *Execute* the ADJUST operation.

Suggestive Response We use a template-based approach (Raux et al., 2005) to generate system responses. Since one of our goals is to ease the learning curve, we includes suggestions or options in the templates. These suggestions aim to guide the user through the image editing process. For example, we show the value range "(-100 to 100)" when asking for VALUE. Figure 5 shows an example of our policy and suggestive response.

5.3. Vision Engine

VISION ENGINE is called when DIALOGUE MANAGER decides to *Query*. The main function of VISION ENGINE is to infer a localized image region from user's referring expression. It acts as an API endpoint for the system, taking image and REFER as input and returns a MASK that outlines the referred object. In our system, we return only the MASK with the highest confidence score.

We use MattNet (Yu et al., 2018) as our system's VISION ENGINE. MattNet is a modular network that decomposes referring expressions into subject appearance, location, and object relationship components. It then uses language and visual attention to infer the referred object in the image. For consistency with NLU, we used MattNet weights trained on RefCOCO (Section 4.).

5.4. Image Edit Engine

IMAGE EDIT ENGINE displays the image and MASK to the user, and performs edits when DIALOGUE MANAGER decides to *Execute* an edit operation. We developed our own Image Edit Engine using OpenCV (Bradski, 2000), which is sufficient for our use cases. Our IMAGE EDIT ENGINE supports the ATTRIBUTE adjustments listed in Section 3., and can highlight particular regions given a MASK.

5.5. Interface

Users interact with our system through a web-based interface. To simulate an image editing software tool, our interface displays the image, with a highlighted region if MASK is provided. We display the supported list of attributes (Section 3.) and sliders. We also show the currently tracked REFER to users so that they can see the result of their input. Natural language is the only way to enter IERs; users are restricted to text inputs (i.e. sliders are not functional).

6. Evaluation

6.1. Setup

To simulate a real world scenario, we used the test images from our NLU dataset (Section 4.), so that both NLU and VISION ENGINE would be exposed to unseen data. We assigned a single AMT user to each image, so each task is done by a different user. We released our code and data on Github. [1]

Instructions We introduced our system as "an image editing chatbot that is able to detect objects in the image and adjust several image attributes". We asked users to perform at least 2 edits and interact with our system for at least 10 turns. There were no restrictions on what to edit. To limit overall dialogue length, users could end the dialogue at 30 turns. Upon task completion, we asked users to rate overall system performance, plus 3 specific system features. We then asked users to provide feedback on what they liked, disliked and what improvements would make the system better.

6.2. Results

Edits We assess our system by the number of edits users can complete in each dialogue. All 83 users completed the required 2 edits, with 5 users exceeded 30 turns. 25 users (30.1%) performed 3 or more edits, with the maximum observed being 5. This implies that our system is usable, and

[1] https://github.com/tzuhsial/ImageEditingWithDialogue

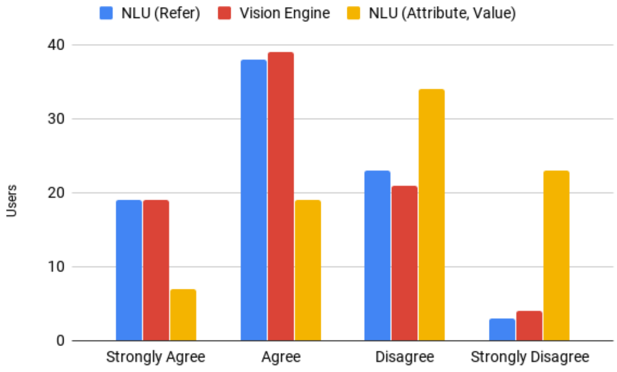


Figure 3: Ratings of NLU-REFER (Blue), VISION ENGINE (Red), and NLU-ATTRIBUTE/VALUE (Yellow). (Strongly) Agree/Disagree were chosen according to a statement "I found it difficult for the chatbot to use `<feature>`". `<feature>` is the system feature description.

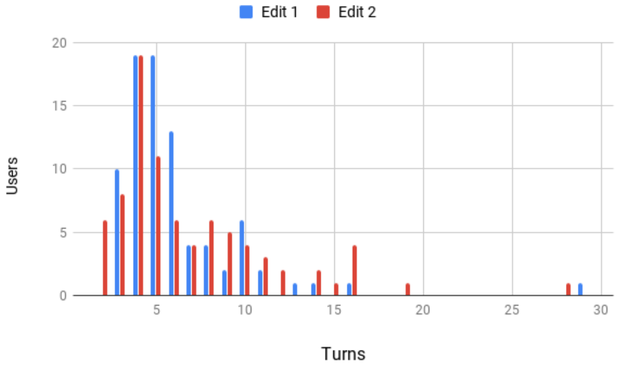


Figure 4: Number of turns to perform the 1st edit (Blue) and the 2nd edit (Red).

engaged users enough for them to have done more than the minimum required.

Ratings Performance and feature ratings are shown in Table 2 and Figure 3. Performance ratings are rated on a 5-point Likert Scale from *Very Poor*, *Poor*, *Fair*, *Good*, to *Excellent*. *Good* received the most ratings (29), and few users (10) rated our system as *Very Poor* or *Excellent*. For system features, we choose NLU-REFER (NLU performance on REFER), VISION ENGINE (Object segmentation), NLU-ATTRIBUTE/VALUE (NLU performance on ATTRIBUTE and VALUE) for rating. ATTRIBUTE and VALUE were combined because of similar preprocessing. According to a statement "I found it difficult for the chatbot to `<feature>`", where `<feature>` is a system feature description, system features are rated on a 4-point Likert Scale, from *Strongly Disagree*, *Disagree*, *Agree*, to *Strongly Agree*. Most users agreed the system had difficulty tracking REFER and inferring the correct MASK (VISION ENGINE). Most users agreed that the system could correctly track ATTRIBUTE and VALUE.

7. Analysis and Discussion

Statistics We collected 83 dialogues with 1359 user utterances (2,753 tokens, 534 unique). The mean number of turns in a dialogue is 17.4; it took an average of 6.3 turns to make an edit.

7.1. Quantitative Analysis

Turns Our initial conjecture was that after performing the 1st edit, users would become more familiar with our system and would need fewer of turns. However, after plotting the turn count the 1st and 2nd edits (Figure 4), we found that the turn distributions were very similar. A Kolmogorov-Smirnov test (Massey Jr, 1951) gives a p-value of 0.93, indicating sampling from the same distribution. While this suggests that our system does not have a learning curve, this phenomena is likely a result of our STATE UPDATER's clearing all slot values after execution. Users may want to edit the same object, but have to select MASK again. We leave slot carry-over (Naik et al., 2018) for future work.

NLU Coverage To understand NLU performance in actual dialogue, we annotated dialogue utterances and computed the F1 scores (Table 1). Though ATTRIBUTE and VALUE retained a F1 score of over 90%, ACTION, REFER dropped from 96% to 63%. This coincides with the system feature ratings from the users, which gave a better rating for NLU-ATTRIBUTE/VALUE than for NLU-REFER.

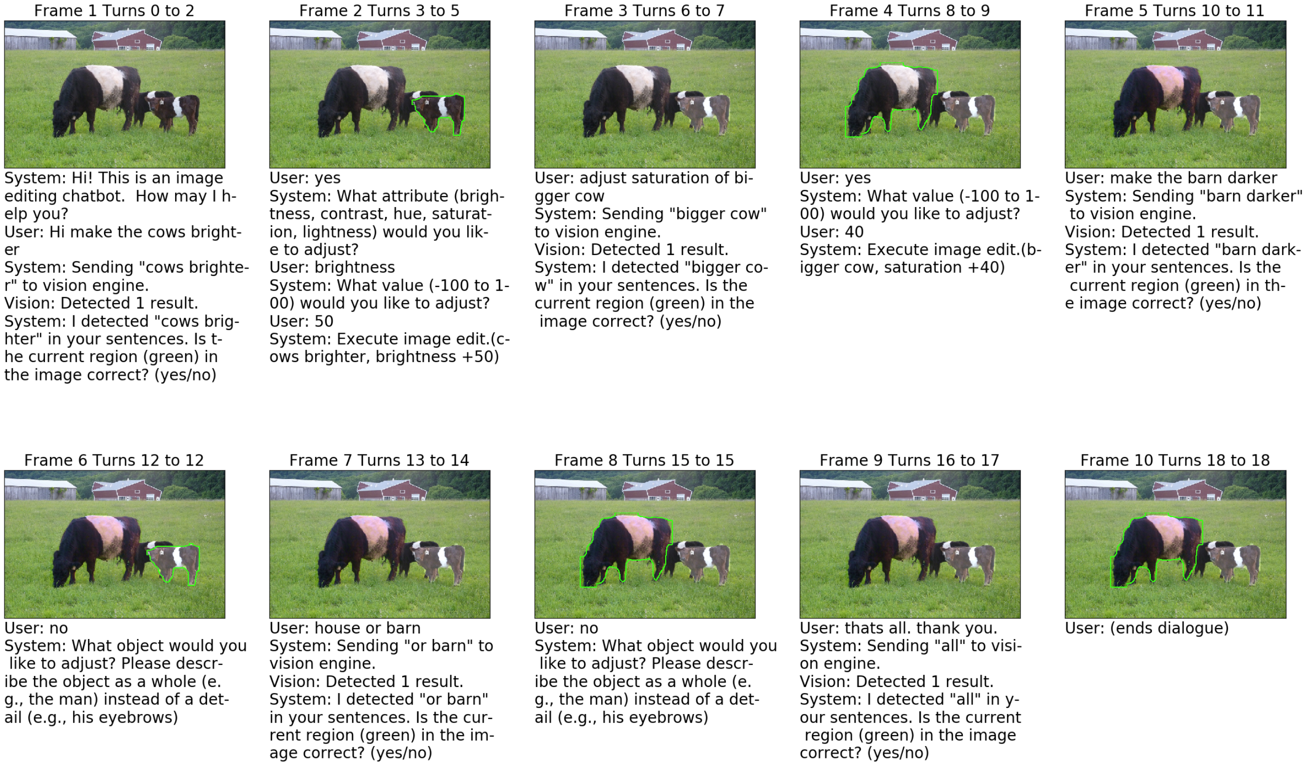
Vision Accuracy Object segmentation is important for image editing, and is usually evaluated by Intersection over Union (IoU) with a fixed threshold (Lin et al., 2014). Nonetheless, users have many reasons to accept/reject a MASK, regardless of IoU. To assess how MASK affects user's ratings, we propose a simple metric to evaluate object segmentation accuracy for image editing systems, based on dialogue actions.

$$
\begin{aligned}
&Vision\ Accuracy = \frac{\#Execute}{\#Query} \\
&0 \le \#Execute \le \#Query
\end{aligned}
\tag{1}
$$

Vision Accuracy divides $\#Execute$ (number of *Execute*) by $\#Query$ (number of *Query*). Since DIALOGUE MANAGER will always *Confirm* MASK, an edit (*Execute*) will only contain a MASK users deem acceptable. $\#Query$ can therefore be thought as the number of attempts to get a better MASK, with $\#Execute$ being the number of acceptances. *Vision Accuracy* represents the object segmentation performance of the whole system rather than just VISION ENGINE, since it depends on NLU and our policy.
We computed a Pearson correlation between *Vision Accuracy* and user ratings. We found that *Vision Accuracy* has a higher correlation with system performance (0.46) than with VISION ENGINE ratings (0.32). This suggests that our proposed metric, though initially conceived to capture VISION ENGINE ratings, could also serve as an indicator of system performance.

7.2. User Feedback

Likes After examining user likes, we classified them into 5 categories: (i) Easy to use (ii) Quick (iii) Capable (iv) Experience (v) Other. For (i), feedback that mentioned the



System: Hi! This is an image editing chatbot. How may I help you?
User: Hi make the cows brighter
System: Sending "cows brighter" to vision engine.
Vision: Detected 1 result.
System: I detected "cows brighter" in your sentences. Is the current region (green) in the image correct? (yes/no)

User: yes
System: What attribute (brightness, contrast, hue, saturation, lightness) would you like to adjust?
User: brightness
System: What value (-100 to 100) would you like to adjust?
User: 50
System: Execute image edit.(cows brighter, brightness +50)

User: adjust saturation of bigger cow
System: Sending "bigger cow" to vision engine.
Vision: Detected 1 result.
System: I detected "bigger cow" in your sentences. Is the current region (green) in the image correct? (yes/no)

User: yes
System: What value (-100 to 100) would you like to adjust?
User: 40
System: Execute image edit.(bigger cow, saturation +40)

User: make the barn darker
System: Sending "barn darker" to vision engine.
Vision: Detected 1 result.
System: I detected "barn darker" in your sentences. Is the current region (green) in the image correct? (yes/no)

User: no
System: What object would you like to adjust? Please describe the object as a whole (e.g., the man) instead of a detail (e.g., his eyebrows)

User: house or barn
System: Sending "or barn" to vision engine.
Vision: Detected 1 result.
System: I detected "or barn" in your sentences. Is the current region (green) in the image correct? (yes/no)

User: no
System: What object would you like to adjust? Please describe the object as a whole (e.g., the man) instead of a detail (e.g., his eyebrows)

User: thats all. thank you.
System: Sending "all" to vision engine.
Vision: Detected 1 result.
System: I detected "all" in your sentences. Is the current region (green) in the image correct? (yes/no)

User: (ends dialogue)

Figure 5: An example dialogue from our user study (Section 6.). At the beginning of the dialogue, user expressed a high-level IER "Hi make the cows brighter". Our system *Confirms* the MASK then *Requests* low-level edit arguments ATTRIBUTE, VALUE from the user. After the 1st edit, user accommodated to our system and expressed an low-level IER "adjust saturation of bigger cow" (Frame 3 Turns 6 to 7). After the 2nd edit, user tried to select the barn "make the barn darker" (Frame 5 Turns 10 to 11), and paraphrased "house or barn" (Frames 7 Turns 13 to 14) when our VISION ENGINE failed to detect the correct object. Finally, user relents and ends the session.

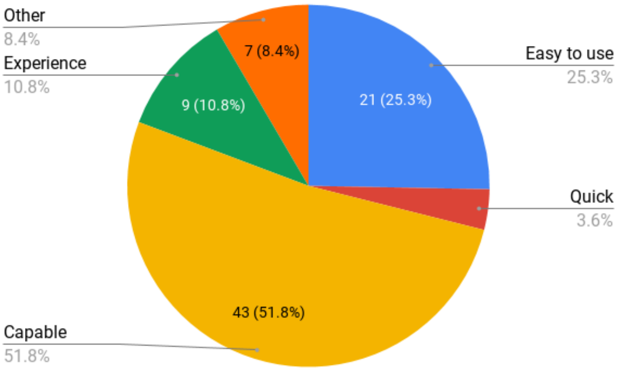


Figure 6: User feedback on what they liked about our system. We divided into 5 categories (i) Easy to use (Blue), (ii) Quick (Red), (iii) Capable (Yellow), (iv) Experience (Green) (v) Other (Orange).

system was easy to use (e.g., "The edit process seemed straightforward", "I liked the ease of use") or mentioned the system actions *Request* or *Confirm* (e.g., "it let you know what it wanted", "It would ask clarifying questions"). For (ii), we included feedback that mentioned our system was quick. For (iii), feedback that mentioned the system was able to understand user utterances, automatically select objects in the image, or perform edits (e.g., "It understood what I said", "Auto select of the area"). For (iv), feedback that praised the whole experience. All other feedback went to (v).

This is summarized in Figure 6: users generally liked the concept of an NLIE system (Capable: 51.8%), and that dialogue created a smooth experience (Easy to use: 25.3%, Experience: 10.8%).

Suggestions For dislikes and suggestions, most users first named a thing they disliked, and then suggested improving it. We report only suggestions from the users.

Most suggestions were centered around object segmentation. 38 users (46%) wanted more accurate MASKS, with several mentioning that they want the system to show a list of objects it could recognize beforehand. Several other users suggested including selection tools to modify the predicted MASK or just select the object themselves.

Another trend we observed what users cared about was speed. 9 users (10.8%) suggested using a better NLU, others wanted the system to stop repeating the same mistakes, implying that we should design or learn a better policy (Williams and Young, 2007). For the interface, some suggested that the system should let users directly modify ATTRIBUTE/VALUE instead text inputs. One user suggested speech recognition, stating that it would be more efficient to speak and edit at the same time. Also, another 9 users (10.8%) suggested that responses be made more human-like.

Overall users indicated acceptance if not preference for a language-based interface; other suggestions were for improvements in the language channel. Users seemed to expect that a language interface would somehow be more intelligent and would better understand the task. We believe this is a critical requirement for such interfaces.

7.3. Discussion

One of our original goals was to ease the learning curve for image editing tools. From our evaluation and analysis, dialogue actions that guide users through the task (one slot at a time) is an effective way to do this. The ability to use natural language descriptions for object selection eases overall effort.

The greatest limitation of our current system is that the range of regions that could be selected is bounded by VISION ENGINE capabilities. As shown in Figure 5, users typically relent after paraphrasing REFER multiple times. Most object segmentation models (Lin et al., 2014) are restricted to detecting common objects, while users may actually want to modify less common objects or non-object regions. A possible solution is to include selection tools and the option to modify system's inferred MASK; region selection is a very delicate procedure. Users should be able to enjoy the convenience of using referring expressions and fall back to manual selection whenever needed. It could be useful to have a learning system that tries to relate region selection to language.

Apart from object segmentation, another limitation is that users would have to express ATTRIBUTE and VALUE using natural language. Initially users are unfamiliar with the jargon and the nature of effects; the dialogue manager should suggest one of the possible ATTRIBUTES. However, as users learn the ATTRIBUTES and how they perform, they would know where the ATTRIBUTE sliders are and preferably manipulate the sliders themselves.

To summarize, for a task like image editing which exhibits a steep learning curve, dialogue systems can help users get familiar with a complex tool. However, once users familiarize with the features, they would choose to interact with natural language if it is easier than performing edit operations themselves (Rudnicky, 1993). Understanding the characteristics of this threshold should shed more light on the design of dialogue systems for tools that have steep learning curves. Additionally, it would be interesting to compare against a high-level language based system (Wang et al., 2018) and see under which circumstances would the user prefer. [2]

8. Conclusion

Image editing is difficult and takes time to master. With the goal of improving the learning curve, we developed a low-level task-oriented dialogue system for image adjustments. Even though our system has only one edit type and requires users to type low-level edit arguments, we find that users take well to a language-based dialogue interface, notable for an essentially visual task. Our work demonstrates the potential of grounding image editing dialogues at low-level, and our analysis shows important aspects for user satisfaction.

For future work, we plan to incorporate additional edit types and develop a multimodal setting that includes gesture manipulations; we expect that language will provide an easy-to-learn interface for users, one that remains available in later use. With the expanded system, we plan to collect a dialogue corpus for dialogue state tracking and training end-to-end models.

9. Bibliographical References

Bohus, D. and Rudnicky, A. I. (2009). The ravenclaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3):332–361.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools.*

Brixey, J., Manuvinakurike, R. R., Le, N., Lai, T., Chang, W., and Bui, T. (2018). A system for automated image editing from natural language commands. *CoRR*, abs/1812.01083.

Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gasic, M. (2018). Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Bychkovsky, V., Paris, S., Chan, E., and Durand, F. (2011). Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE 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*, pages 8721–8729.

Cheng, Y., Gan, Z., Li, Y., Liu, J., and Gao, J. (2018). Sequential attention gan for interactive image editing via dialogue. *arXiv preprint arXiv:1812.08352.*

Dale, R. and Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*, 19(2):233–263.

Dhingra, B., Li, L., Li, X., Gao, J., Chen, Y.-N., Ahmed, F., and Deng, L. (2017). Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–495.

El Asri, L., Schulz, H., Sharma, S., Zumer, J., Harris, J., Fine, E., Mehrotra, R., and Suleman, K. (2017). Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219.

Foley, J. D., Van, F. D., Van Dam, A., Feiner, S. K., Hughes, J. F., HUGHES, J., and ANGEL, E. (1996). *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio,

[2] According to the original authors, their data was lost and thus their work could not be compared.

Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

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

Jhamtani, H. and Berg-Kirkpatrick, T. (2018). Learning to describe differences between pairs of similar images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4024–4034.

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)*, pages 787–798.

Krahmer, E. and Van Deemter, K. (2012). Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

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*, pages 740–755. Springer.

Lin, T.-H., Bui, T., Kim, D. S., and Oh, J. (2018). A multimodal dialogue system for conversational image editing. *NeurIPS 2018 2nd Conversational AI Workshop: Today's Practice and Tomorrow's Potential*.

Manuvinakurike, R., Brixey, J., Bui, T., Chang, W., Kim, D. S., Artstein, R., and Georgila, K. (2018a). 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. (2018b). Conversational image editing: Incremental intent identification in a new dialogue task. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 284–295.

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

Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Mohapatra, A. (2018). Natural language driven image edits using a semantic image manipulation language. Master's thesis, Virginia Tech.

Naik, C., Gupta, A., Ge, H., Lambert, M., and Sarikaya, R. (2018). Contextual slot carryover for disparate schemas. *Proc. Interspeech 2018*, pages 596–600.

Raux, A., Langner, B., Bohus, D., Black, A. W., and Eskenazi, M. (2005). Let's go public! taking a spoken dialog system to the real world. In *Ninth European conference on speech communication and technology*.

Rudnicky, A. I. (1993). Factors affecting choice of speech over keyboard and mouse in a simple data-retrieval task. In *Third European Conference on Speech Communication and Technology*.

Suhr, A., Zhou, S., Zhang, A., Zhang, I., Bai, H., and Artzi, Y. (2019). A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Tan, H., Dernoncourt, F., Lin, Z., Bui, T., and Bansal, M. (2019). Expressing visual relationships via language. *arXiv preprint arXiv:1906.07689*.

Wang, H., Williams, J. D., and Kang, S. (2018). Learning to globally edit images with textual description. *arXiv preprint arXiv:1810.05786*.

Wen, T., Vandyke, D., Mrkšíc, N., Gašíc, M., Rojas-Barahona, L., Su, P., Ultes, S., and Young, S. (2017). A network-based end-to-end trainable task-oriented dialogue system. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017-Proceedings of Conference*, volume 1, pages 438–449.

Williams, J. D. and Young, S. (2007). Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

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*, pages 1307–1315.