Abstract

Text-based image retrieval has seen considerable progress in recent years. However, the performance of existing methods suffers in real life since the user is likely to provide an incomplete description of an image, which often leads to results filled with false positives that fit the incomplete description. In this work, we introduce the partial-query problem and extensively analyze its influence on text-based image retrieval. Previous interactive methods tackle the problem by passively receiving users’ feedback to supplement the incomplete query iteratively, which is time-consuming and requires heavy user effort. Instead, we propose a novel retrieval framework that conducts the interactive process in an Ask-and-Confirm fashion, where AI actively searches for discriminative details missing in the current query, and users only need to confirm AI’s proposal. Specifically, we propose an object-based interaction to make the interactive retrieval more user-friendly and present a reinforcement-learning-based policy to search for discriminative objects. Furthermore, since fully-supervised training is often infeasible due to the difficulty of obtaining human-machine dialog data, we present a weakly-supervised training strategy that needs no human-annotated dialogs other than a text-image dataset. Experiments show that our framework significantly improves the performance of text-based image retrieval. Code is available at https://github.com/CuthbertCai/Ask-Confirm.

1. Introduction

Recently, cross-modal retrieval, especially text-based image retrieval has gained increasing attention [36]. Although significant improvement has been achieved with existing methods [15, 36, 7] for text-based retrieval, we found in practice their retrieval result is barely satisfactory when users only describe some local regions in an image.

In this work, we introduce a new concept of partial-query problem in text-based image retrieval, where the initial text query only describes some objects in the target image. Studies [30, 28] have found that when examining an image, people tend to only focus on the objects that stand out the most. This could lead to problems where the objects that people focus on are not the discriminative objects that can distinguish the target image from similar candidates, thus making the user’s input insufficient for retrieving the target image. As shown in Figure 2 (a) and (b), a cross-modal retrieval model performs poorly when a query is only partially given. In both examples, the target image ranks lower than 1000th, while the other false positives rank top three. A common object (blue box) described by the partial query is presented in all images. However, the rest of images are vastly different. For example, in Figure 2 (a), besides the stroller mentioned in the query, the target image consists of umbrellas, chairs, and so on. Whereas the others consist of different objects like trees and buses. If the retrieval model receives a complete description including all objects, existing methods [15, 17, 32] perform excellently.

To show how the partial query hurts retrieval, we test two
text-image retrieval models, S-SCAN and T-CMPL on Visual Genome [14], which are modified from SCAN [15] and CMPL [36]. The implementation is detailed in Section 4. For each image, its complete description includes 10 captions for different regions. We gradually decrease the number of captions and use them as queries to retrieve the target image. As shown in Figure 2 (c) and (d), for both models, R@10 decreases and Mean Rank increases as the degree of incompleteness increases. These results reveal that partial queries should be tackled for a robust retrieval model.

Existing interactive retrieval models [6, 33, 26, 13, 11, 35, 7, 31] tackle the partial query by involving feedback of users in the retrieval process. Given the initial queries from users, these methods first give several relevant candidates that could potentially be the target image. By comparing the target image with these reference images, users give the retrieval method different forms of feedback to describe the difference between them, such as scores [26, 33], tags [13, 11, 12] or descriptions [7, 29]. The models then refine the retrieval results according to the user feedback and continue next round of iteration until the target image is found. Previous methods only passively receive additional information from users, so users need to have substantial practice and expert knowledge on the retrieval system to give discriminative feedback that can quickly narrow down the retrieval range. Hence, to free users from the burden of analyzing the retrieval results and looking for the discriminative information, we propose that the retrieval model itself should be able to actively search for the discriminative information the current query misses. Another problem of previous interactive retrieval models is time-consuming. For example, description-based methods [7, 29] require users to input long sentence feedback and tag-based methods [13, 11, 12] require users to input a bunch of attributes. Hence, we propose a framework where users only need to make simple yes/no confirmation on AI’s question.

In this paper, we propose a novel interactive retrieval framework called Ask&Confirm as shown in Figure 1. The agent first retrieves a set of relevant candidates from the gallery based on initial text queries. Then, it will analyze the retrieval results and the overall status of gallery, and actively select discriminative object candidates for users to confirm their presence. Based on users’ confirmation, the agent narrows down the range of candidates and eventually gathers enough information to locate the target image. Instead of passively receiving user feedback, a reinforcement learning (RL) based policy is trained to actively search for the discriminative objects missed in the query, and use these objects to distinguish the target image from the rest of gallery. In this active object-based interaction, users only need to confirm the existence of the proposed objects in the target image, no expert knowledge on the retrieval task and extra effort is needed. Moreover, unlike previous RL based interactive methods [7, 19] that require human-annotated dialogs which is impractical to widely collect, our Ask&Confirm framework is trained in a weakly-supervised manner, where only text-image pairs are needed.

The contributions of our framework are as follows: 1) To our knowledge, this is the first work that formally addresses and analyzes the problem of partial query in cross-modal retrieval. 2) Instead of passively receiving missing details from user feedback, we propose a novel interactive retrieval framework Ask&Confirm that introduces an active object-based interaction to actively select the most discriminative objects for users to confirm. 3) Rather than using human-annotated dialogs, we propose a weakly-supervised reinforcement learning framework to optimize the interactive policy that explores the statistical characteristics of the gallery. Experiments show that our framework is effective and robust with partial queries.

2. Related Work

2.1. Text-based Image Retrieval

Most text-based image retrieval approaches are based on deep neural networks [36, 15, 17, 9, 32, 5]. The objective of them is to accurately measure the similarity between the inputs from two different modalities. Cross-Modal Projection Learning (CMPL) [36] is proposed to pull image and text embeddings into an aligned space. To further enhance the retrieval in a fine-grained way, [15, 17, 9, 32] proposed different attention-based approaches, applying visual attention between every image region and word.
2.2. Query Expansion

Query expansion tackles incomplete information. Different from partial queries that are complete sentences of local regions, it focuses on queries that are incomplete sentences. An incomplete sentence as the query leads to poor retrieval. Thus, query expansion methods are proposed [37, 18, 22, 4, 8]. [37] learns users’ searching history to generate expansion. [18] explores expansion by calculating similarity distance in thesaurus indexed collections. Other methods [22, 4, 8] that focus on image or video retrieval provide expansion based on knowledge bases.

2.3. Visual Dialog

Visual dialog aims to let the machine understand the visual content and have a natural conversation with the user about it. After examining the image, the agent can answer the user’s questions on different aspects. Mainstream approaches are based on policy-based reinforcement learning to achieve good question-answer performance [24, 2, 3]. However, the dialogs are purely text-based for both the questioner and answer agent, and a manually annotated dialog dataset is needed to train a visual dialog system.

2.4. Interactive Image Retrieval

The retrieval model is hard to locate the target image with the initial query. Inspired by visual dialog, interactive image retrieval systems [33, 26, 13, 12, 11, 25, 16, 20, 21] are proposed to solve this problem. In these systems, users give feedback to an agent according to a reference image. There are two types of feedback: relevance and difference. For the former one [33, 26], users give relevance scores for the current retrieval results. Then the system re-ranks its retrieval results by using the user’s feedback. For the latter one [13, 12, 11, 25, 34], users tell the difference between the target image and a reference image to the system with tags or descriptions. The system then whittles away the irrelevant images and ranks the correct one to the top.

3. Method

3.1. Object-based Interaction

In a partial-query problem, one of the most important tasks for an interactive retrieval model is to obtain the missing discriminative information that can distinguish the target image from others. Generally, the demands of more discriminative information and less user effort are contradictory, because more information usually means that the user has to pay more effort to think about what is the most discriminative thing and to input more descriptions. For example, tag-based methods [13, 11, 12] only require the user to point out a different attribute between the target image and a reference image, but they hardly filter out many negative

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
<th>MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-SCAN</td>
<td>4.5</td>
<td>13.6</td>
<td>20.4</td>
<td>416</td>
</tr>
<tr>
<td>S-SCAN+Objects</td>
<td>46.4</td>
<td>70.2</td>
<td>78.4</td>
<td>28.4</td>
</tr>
</tbody>
</table>

Table 1: Retrieval improvements over S-SCAN with ground-truth object descriptions. MR means Mean Rank.

images per round because too little discriminative information is provided. On the contrary, description-based methods [7, 29] require the user to give long sentence feedback that enriches more details but pays more user effort.

In Ask&Confirm, we propose an object-based interaction where a RL-based policy actively searches for discriminative object candidates for users to confirm, then users just need to confirm whether objects are in the target image. Under the auxiliary of the active policy, the demands of more discriminative information and less user effort are simultaneously satisfied.

We choose object-based interaction based on two main reasons: (1) objects in an image are discriminative enough to distinguish different images, (2) objects can be easily obtained with a pre-trained detector such as RCNN [1].

Firstly, we discover that the distribution of objects in an image gallery is generally low-entropy, making it a discriminative feature for retrieving the target image. For example, in Visual Genome [14], some objects, such as “trophy” and “skateboard”, rarely appear. If an image includes them, they are discriminative enough to narrow down the retrieval scope quickly. To verify this observation, two types of queries are compared using the same retrieval method S-SCAN: partial query only and supplement partial query with the name of the objects. As shown in Table 1, remarkable improvements are achieved by adding object words, verifying that objects contain discriminative information to distinguish the target image from the rest of the gallery.

Second, the convenience of obtaining objects of an image also makes the object-based interaction practical. Previous text-based image retrieval methods [5, 15, 17, 32] extract image features by an object detector [1]. By reusing the detector, we can directly obtain objects of each image.

3.2. Interactive Retrieval Agent

By adopting the proposed object-based interaction, we propose an interactive retrieval agent to tackle the partial-query problem. It takes the charge of extracting features, interacting with the user and retrieving the target image. In this section, we illustrate how the agent works, especially how it actively searches for object candidates for the user to confirm, which greatly reduces the user effort.

Define a set of captions $Q = \{q_i\}_{N=1}^{N}$ that composes descriptions of an image $i$, where each $q_i$ describes a region. By regarding $Q$ as queries, the goal of a retrieval agent $R$ is to retrieve the target image $i$, from a gallery $T = \{i_n\}_{n=1}^{N}$ through $T$ rounds interaction with the user.
The partial-query problem considers that $Q$ only describes parts of an image instead of the full image.

The interactive retrieval agent $R$ includes four main components: Text Encoder, Image Encoder, Candidate Generator and Ranker. As the interactive workflow illustrated, Text Encoder and Image Encoder embed partial queries and images to a textual-visual feature space as textual features and visual features respectively. At each round, Candidate Generator actively searches for the most discriminative objects as candidates for the user to confirm. Given the objects, the user confirms them as the positive or negative, where positive objects refer to the ones that exist in the target image, vice versa. Then, names of positive objects are added to the partial query and the new query’s feature is updated by the Text Encoder. Finally, based on positive objects, negative objects and the features of queries and images, Ranker retrieves the target image. In detail, Ranker first computes an initial similarity between the textual query and visual features. Secondly, the initial similarity is further refined by the user-confirmed objects. If a gallery image contains the negative objects, the similarity between the image and queries would be refined to a lower value. The retrieval result of the current round is given by the refined similarity. Below we provide details on the specific design of each component.

**Text Encoder.** At $t$ th round, the input partial queries are denoted as $Q_t = \{q_n\}_{n=1}^{N^Q}$. They are embedded into textual features by Text Encoder (TE):

$$x_n^T = T E(q_n), q_n \in Q_t$$

where $x_n^T$ denotes a texture feature. The set of all textual features of $Q_t$ is denoted as $X^T_t = \{x_n^T\}_{n=1}^{N^Q}$. In detail, we use a gated recurrent unit as $TE$ just like [29].

**Image Encoder.** Given an image gallery $I = \{i_n\}_{n=1}^{N}$, Image Encoder (IE) extracts the visual feature and detects objects for each image:

$$(x_n^I, A_n) = I E(i_n)$$

where $x_n^I$ denotes the visual feature of $i_n$ and $A_n$ denotes the objects $\{a_1, a_2, \ldots\}$ that appear in $i_n$. The set of all visual features of $I$ is denoted as $X^I = \{x_n^I\}_{n=1}^{N}$.

**Candidate Generator.** At $t$ th round, Candidate Generator actively searches for the most discriminative objects as candidates for the user to confirm positive objects that appear in the target image $i_t$. These candidates are denoted as $A_t = \{a_n\}_{n=1}^{N_A}$. The user confirms positive objects $A^+_t = \{a^+_n\}_{n=1}^{N^+_A}$, thus, the rest of $A_t$ are negative objects. They are denoted as $A^+_t = \{a^+_n\}_{n=1}^{N^+_A}$ where $N^+_A + N^A_A = N_A$.

The text of $A^+_t$ is used as the additional description of $i_t$. It is denoted as $Q^+_t = \{\mathcal{T}(a^+_n)\}_{n=1}^{N^+_A}$, where $\mathcal{T}(a^+_n)$ is the word of $a^+_n$.

**Ranker.** Given $X^T_t$, $X^I_t$, and $A^+_t$, Ranker gives a retrieval result of $t$ th round. Firstly, Ranker computes the similarity between queries and each image, where $S_{t,n}(X^T_t, x_n^I)$ denotes the similarity between $X^T_t$ and $x_n^I$. Secondly, if $x_n$ contains negative objects belong to $A^+_t$, we refine $S_{t,n}$ with a lower value where $S_{t,n} := S_{t,n} \times 0.9$. With the refined similarity, Ranker gives a retrieval result.

### 3.3. Weakly-supervised Policy Learning

The key of Ask&Confirm to satisfy the demand of more discriminative information and less user effort is an active search policy. It selects the most discriminative objects as candidates for users to confirm, according to the textual feature and the object distribution of an image gallery. Thus, it frees the user to think about what are the most discriminative objects and inputs long sentence feedback.

In this work, the active search policy is learned with a weakly-supervised RL-based training. The weakly-supervised policy learning automatically finds an optimal policy by letting the agent iteratively interact with users and self-update based on the users’ feedback. The whole policy learning is very concise and can be easily conducted in a weakly-supervised manner, because we only need to know objects in each image and users’ feedback can be mimicked by ground-truth objects in the target image. We can even just reuse the detector for extracting image features to detect objects. On the contrary, previous dialog-based retrieval methods [7, 19, 3] require burdensome collections of chatting sessions. The superiority that our method needs no extra data collections makes it more practical.

**Reinforcement Learning.** The policy obtained by Candidate Generator is modeled as a policy $\pi$, parameterized with $\phi_\pi$, which outputs each object’s probability $P(a)$ of getting selected. The five components in our policy learning action, state, policy, value and reward are as follows:

- **Actions** refers to the objects selected by Candidate Generator at each round, i.e., $a \in A$, $A$ is the set of all objects.

- **State** $s^t$ is defined as a concatenation of $s^t_1 = \sum_{n=1}^{N^+_A} \frac{x_n^I}{N^Q}$ and $s^t_2 = P_r(a)$, where $P_r(a)$ is the distribution of $a$ among the top 100 images generated by the Ranker. We utilize such design to make $\pi$ aware of information both from partial queries and the ranking list.

- **Reward** is defined as the similarity between textual features and the target visual feature, i.e., $S(X^T_t, x^I_t)$ where $x^I_t$ is the visual feature of the target image.

**Policy $\pi$** is implemented with a three-layer MLP. The object sampling distribution $P(a)$ is approximated with $\pi(s^t)$.

**Value** is estimated with $V(s^t)$. The value net $V$ is implemented with a two-layer MLP parameterized with $\phi_v$.

Given the actions, state, reward, value and policy, a Proximal Policy Optimization (PPO) [27] is applied to optimize the policy net $\pi$ and $\phi_v$. Please refer to the original paper of PPO for more details.
Shaping. RL is hard to converge for dialog agents [23, 7], thus, previous RL-based dialog agents [7, 24, 3] adopt a supervised learning with annotated dialogs for shaping the RL training. To avoid the burdensome human annotation, we propose a weakly-supervised shaping method without annotated dialogs. Our motivation is that objects in the target image should have a high probability to be selected, because adding these objects into the queries could potentially significantly increase the similarity between queries and the target image. However, this probability is infeasible to obtain during test time, because the target image is unknown. As a result, instead of obtaining the probability of objects existing in the target image, we approximate it with the probability of objects that semantically relevant to the target image’s corresponding query. For example, if the target image’s corresponding query is “a man is surfing”, we can infer that objects relevant to this query (e.g., “man”, “sea” and “surfboard”) should have a high probability to appear in the target image. The semantic relevance between an object and a query can be estimated by the conditional probability of an object $a_j$ existing in the query’s corresponding target image, given the query $Q_t$, denoted as $P(a_j|Q_t)$. $P(a_j|Q_t)$ can be estimated by computing the frequency of $a_j$ and $Q_t$ both appearing in the same target image $i_k$:

$$P(a_j|Q_t) = \frac{\sum_{k=1}^{N} \mathbb{1}(a_j \in i_k) \mathbb{1}(Q_t \in i_k)}{\sum_{m=1}^{A} \sum_{k=1}^{N} \mathbb{1}(a_m \in i_k) \mathbb{1}(Q_t \in i_k)}$$

(3)

where $\mathbb{1}(\cdot)$ is an indicator function. $Q_t \in i_k$ is a corresponding query of $i_k$ and $a_j \in i_k$ denotes an object in $i_k$.

A practical problem is that $Q_t$ hardly appears in different $i_k$, which causes $\sum_{k=1}^{K} \mathbb{1}(a_j \in i_k) \mathbb{1}(Q_t \in i_k)$ always being 1. Thus, we use a set of words $\{w_n\}_{n=1}^{N_w}$ to represent $Q_t$, where $w_n$ is a tokenized word in $Q_t$. The tokenized word $w_n$ could appear in different images. $\mathbb{1}(Q_t \in i_k)$ is replaced with $\sum_{n=1}^{N_w} \mathbb{1}(w_n \in i_k)$. $P(a_j|Q_t)$ is then modified to:

$$P(a_j|Q_t) = \frac{\sum_{k=1}^{K} \sum_{n=1}^{N_w} \mathbb{1}(a_j \in i_k) \mathbb{1}(w_n \in i_k)}{\sum_{m=1}^{A} \sum_{k=1}^{K} \sum_{n=1}^{N_w} \mathbb{1}(a_m \in i_k) \mathbb{1}(w_n \in i_k)}$$

(4)

Guiding with $P(a|Q_t)$, we then train $\pi$ by optimizing

$$L_\pi = \sum_{t=1}^{N_t} (P(a(Q_t)) - \pi(s_t))^2$$

(5)

where $N_t$ means that $L_\pi$ is optimized for every $N_t$ rounds.

Combining RL with the shaping, loss of the policy learning process is $L = L_p + \alpha \cdot L_s$, where $L_p$ denotes the loss of PPO and coefficient $\alpha$ is used to balance the RL learning and shaping. The shaping is crucial in our method otherwise the training process cannot converge.

4. Experiments

Dataset. There is no existing benchmark for interactive partial-query retrieval and we build a new dataset based on Visual Genome [14]. In Visual Genome, multiple regions are detected by an object detector [1] for each image, and each of the object region is annotated with a description. We preprocess the data by following the protocol in [29], resulting in 105,414 images. Images are split into 92,105/5,000/9,896 for training/validation/testing. To perform an interactive partial-query retrieval without extra data collection, we regard a region caption as a partial query offered by users and objects in the target image as feedback from users. All evaluations are performed on the test split.

Baselines. Ask&Confirm is a simple framework compatible to any cross-modal retrieval methods. We implement variants of SCAN [15] and CMPL [36], which are named Simplified SCAN (S-SCAN) and CMPL with Triplet loss (T-CMPL) respectively, as the basic retrieval models and build the proposed interactive retrieval agent on them. Both of the variants adopt the text and image encoder in Section 3.2 to obtain textual features $X^T = \{x^T_j\}_{j=1}^{A}$ and visual features $X^I = \{x^I_{k,m}\}_{k,m=1}^{K,M}$. (a) S-SCAN: We modify the bidirectional attention mechanism in SCAN to a unidirectional one to adopt multi-query inputs. Thus, the similarity between $x^T_j$ and $x^I_k$ is modified as

$$S_{j,k}(x^T_j, x^I_k) = \frac{1}{M} \sum_{m=1}^{M} \gamma_{j,k} \cdot \cos(x^T_j, x^I_{k,m})$$

(6)
Table 2: Results of Ask&Confirm on S-SCAN after 10 rounds. AC denotes the Ask&Confirm framework.

\[
\gamma_{j,k} = \exp(\cos(x_j^T, x_k^T)) / \sum_{m=1}^{M} \exp(\cos(x_j^T, x_m^T)) \quad \text{and} \quad \cos \text{ denotes the cosine similarity.}
\]

The similarity between \(X^T\) and \(x_k^T\) is the average of \(S_{j,k}\) among all \(x_j^T\).

Both S-SCAN and T-CMPL are optimized with a common ranking loss. It is clear that Ask&Confirm focuses on the interactive mode and is independent of the network architecture and similarity computing. Thus, Ask&Confirm can adopt any existing cross-modal retrieval models.

### Implementation Details
During training, \(T\) is set to 20 to conduct twenty-round interaction. In each round, we set \(N_A = 10\) which means sampling 10 objects from the object sampling distribution \(P(a)\). During testing, we vary \(T\) and \(N_A\) and apply a greedy sampling to choose objects with the highest probabilities. Similar to [1], we utilize a Faster RCNN pretrained on Visual Genome with 1600 object categories to extract features of the top 36 regions and predict objects of regions. Textual and visual features are mapped into vectors with a dimension of 256. For the optimization of policy learning, we update all parameters for every 600 rounds and adopt Adam [10] as the optimizer. Learning rates of \(\phi_{\pi}\), \(\phi_{e}\) are \(3e^{-4}\) and \(1e^{-3}\). Coefficient \(\alpha\) is set to 1000. All models are trained for 500 epochs.

### Evaluation Metrics
We adopt the common R@K (K=1, 5, 10) metric and Mean Rank (MR) to measure the retrieval performance. R@K indicates the percentage of the queries where at least one ground truth is retrieved among the top-K candidates.

### 4.1. Results
Based on S-SCAN and T-CMPL, we build two interactive retrieval models with the proposed Ask&Confirm framework. To prove the effectiveness of Ask&Confirm, we test them in three settings: (1) Q1/A10, (2) Q2/A5, and (3) Q4/A3. QK means K queries are given by users in the beginning, and AK means K objects are provided by an agent in each round. All results are recorded after 10 rounds.

Results are illustrated in Table 2 and 3. For both basic retrieval models in three different settings, Ask&Confirm strengthens their performance in all evaluation metrics. Ask&Confirm enhances R@10 of S-SCAN from 20.4% to 59.8% and strengthens R@10 of T-CMPL by 20.7% with Q1/A10. In the other two settings, the advantage of R@10 brought from Ask&Confirm recedes a bit but at
### Table 3: Results of Ask&Confirm on T-CMPL after 10 rounds. AC denotes the Ask&Confirm framework.

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
<th>MR</th>
<th>Q</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-CMPL</td>
<td>3.1</td>
<td>10.5</td>
<td>16.3</td>
<td>393.4</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>T-CMPL+AC</td>
<td>5.2</td>
<td>20.4</td>
<td>37.0</td>
<td>313.8</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>T-CMPL</td>
<td>7.3</td>
<td>19.5</td>
<td>28.3</td>
<td>283.3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>T-CMPL+AC</td>
<td>8.6</td>
<td>26.9</td>
<td>47.2</td>
<td>211.3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>T-CMPL</td>
<td>14.5</td>
<td>33.5</td>
<td>44.0</td>
<td>118.2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>T-CMPL+AC</td>
<td>15.1</td>
<td>38.6</td>
<td>59.5</td>
<td>98.7</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

least achieves 14.2%. As for R@5, Ask&Confirm based on S-SCAN achieves 61.4% and the one based on T-CMPL achieves 38.6% with Q4/A3. In other settings, the enhancement of Ask&Confirm is more obvious and even achieves 11.5% with Q2/A5 based on S-SCAN. Both basic retrieval models are improved by Ask&Confirm of R@1 in all settings. In particular, Ask&Confirm based on S-SCAN achieves R@1=34.1% with Q4/A3. With Ask&Confirm, MR of both basic retrieval models in three settings is moved up by a large margin. These results demonstrate the effectiveness of Ask&Confirm.

### 4.2. Visualizations

Examples of interactive retrieval are shown in Figure 4. Several interesting discoveries are found out in visualizations. Firstly, the agent tends to offer several objects in the first few rounds regularly, such as “window”, “man”, “sky”, “head”, “tree” and so on. These are the objects that come up most frequently in Visual Genome. This is a reasonable choice because it either has a large possibility to add a ground-truth object to queries or eliminates plenty of images that include these objects. Secondly, the agent can offer objects that are not common but related to the semantics of given queries and images in later rounds. For example, to retrieve the image that includes zebras, the agent offers “field” and “fence” in round 9 which rarely occur but are related to zebras. To retrieve the image with a query “White short sleeve shirt”, the agent offers “sunglasses” and “top” in round 5 which belong to clothing just like the query, and offers “car” which shows in the image. We ascribe these properties to our policy learning approach. The statistically-based shaping guides the agent to give priority to the most frequent objects and the reinforcement learning promotes objects related to the semantics of images.

### 4.3. Ablation Studies

#### Number of Query and Action

To verify Ask&Confirm is robust to the number of queries and actions, we test it on S-SCAN with different query numbers $N_Q$ ∈ {1, 2, 4} where users input 1, 2, or 4 queries and different action numbers $N_A$ ∈ {3, 5, 10} where the agent provides 3, 5, or 10 object candidates in each round. Results on R@5, R@10, and MR in each round are shown in Figure 5.

In detail, with the same queries, the performance of Ask&Confirm gradually improves when $N_A$ increases, which shows that more actions in each round facilitate the retrieval. On the other hand, when $N_A$ is fixed, queries with higher $N_Q = 4$ outperform the ones with lower $N_Q$, which is consistent with our discovery in Figure 2. Although models with fewer queries achieve worse performance, improvements over them are even more. Especially, when $N_Q = 1$ and $N_A = 10$, Ask&Confirm achieves the largest improvement. We conclude that fewer queries leave more space for the agent to optimize the basic model’s retrieval. Despite the change of the number of queries and actions, Ask&Confirm consistently enhances S-SCAN on all metrics. It examines the robustness of Ask&Confirm which facilitates retrieval stably in all situations.

#### Policy

Finding a policy that guides the agent to choose discriminative objects is essential to Ask&Confirm. As a result, we compare our policy learning method with three pre-defined policies: (1) **Random**: In each round, the agent samples objects from a uniform distribution. (2) **QASim**: Inspired by [18], objects that have similar textual features with a query are preferred. We use cosine similarity between textual features of queries and objects to indicate their similarity. (3) **QACohe**: Considering that some objects tend to occur coherently, such as “building” and “window”, we compute a joint distribution $P_c(a_i, a_j)$ in the train split, where $a_i$ and $a_j$ are in the same image. Then, we use $P_c(a^*, a_j)$ where $a^* = \arg\max_{a \in A} \frac{1}{N_Q} \sum_{n=1}^{N_Q} \cos(x^T_n, TE(T(a)))$ to sample objects.

Experiments based on S-SCAN are conducted in three settings just like Section 4.1. As shown in Figure 6, under all settings, the proposed policy learning outperforms the other by a large margin in terms of R@10. After 10 rounds, our policy learning strategy outperforms the second-best policy by 12.1%, 7.2%, and 5.0% in three settings. We also observe that a good policy increases R@10 rapidly in the first several rounds and slows down in subsequent rounds. Such a policy provides better interactive experiences because users retrieve the target image with fewer interactions.

#### Model Agnostic

By comparing the improvements on S-SCAN and T-CMPL as shown in Table 4, we examine that the proposed framework is model-agnostic. Although the implementation and performance of T-CMPL and S-SCAN are different, Ask&Confirm strengthens both of them on all evaluation metrics. In detail, the two models’ improvements of MR are very close. As for R@K metrics, improvements are more obvious on S-SCAN due to its better original performance. These results demonstrate that Ask&Confirm can easily cooperate with a common text-based retrieval model to boost the retrieval performance.

### 4.4. User Study

To demonstrate the advantage of the active object-based interaction over tag-based and description-based interac-
Table 4: Results of Ask&Confirm on different basic retrieval models after 10 rounds.

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
<th>MR</th>
<th>Q</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-CMPL+AC</td>
<td>+1.9</td>
<td>+39.9</td>
<td>+10.7</td>
<td>-279.6</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>S-SCAN+AC</td>
<td>+4.1</td>
<td>+20.3</td>
<td>+39.4</td>
<td>-320.0</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>T-CMPL+AC</td>
<td>+1.3</td>
<td>+11.5</td>
<td>+8.9</td>
<td>-7.0</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>S-SCAN+AC</td>
<td>+2.1</td>
<td>+26.0</td>
<td>-96.0</td>
<td>2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>T-CMPL+AC</td>
<td>+0.6</td>
<td>+15.2</td>
<td>+14.2</td>
<td>-19.2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>S-SCAN+AC</td>
<td>+0.6</td>
<td>+15.2</td>
<td>+14.2</td>
<td>-21.1</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

As shown in Table 4, we compare Ask&Confirm (AC) with Drill-Down (DD) [29] and WhittleSearch (WS) [13] where DD is a description-based method and WS is a tag-based method. To make a fair comparison of interactive mode, we re-implement DD and WS based on S-SCAN and adopt their interactive mode. 50 images are sampled from the test set. For each image, 4 different users (details in supplementary) are required to retrieve it in 5 rounds with 3 different methods. The retrieval performance in terms of R@1, R@5, R@10 and Mean Rank (Mean) are shown in Figure 7 (a).

To evaluate users’ effort on different methods, we record the average time users take to retrieve each image. AC costs 37.67s, DD costs 53.60s and WS costs 35.18s.

**Conclusion on Performance:** Ask&Confirm achieves similar R@k accuracy and much better Mean Rank compared to DD. Ask&Confirm significantly outperforms WS.

**Conclusion on User Effort:** Ask&Confirm takes significantly less time to complete the retrieval compared to DD and takes similar time compared to WS.

Overall, Ask&Confirm achieves similar performance with description-based interaction and similar retrieval time with tag-based interaction. It examines that Ask&Confirm not only achieves a friendly user experience, but also achieves excellent retrieval performance. Furthermore, Fig.

5. Conclusion

We firstly introduce the partial-query problem that easily makes cross-modal retrieval models collapsed and propose Ask&Confirm, an interactive retrieval framework, to tackle this problem. Ask&Confirm heuristically guides users to enrich details of images by actively searching for discriminative objects of the target image for users to confirm. A weakly-supervised RL-based policy is proposed to conduct the active search, which leverages the characteristics of the image gallery. Experimental results demonstrate the effectiveness and robustness of Ask&Confirm. The weakly-supervised training procedure also makes it more practical than other dialog-based retrieval models.

6. Acknowledgement

This work is supported by Joint Funds of the National Science Foundation of China under Grant U1809206, the Shanghai Municipal Science and Technology Committee of Shanghai Outstanding Academic Leaders Plan under Grant 19XD143400, the Projects of International Cooperation of Shanghai Municipal Science and Technology Committee under Grant 19400720009, the National Natural Science Foundation of China under Grant 61772300, Grant 61771186, Grant 61771164, National Key R&D Program of China under Grant 2020YFA0711400, Shanghai Municipal Science and Technology Major Project (2021SHZDX0101), Shanghai Municipal Commission of Science and Technology Project (195111322001), the Changjiang Scholars Program of China, and the Fundamental Research Funds for the Central Universities.
References


