Local-level Feature Aggregation with Attribute Anchors for Text-Guided Image Retrieval

Chan Hur
School of Electrical Engineering and Computer Science
Kyungpook National University
Daegu, KOREA
chanhur@knu.ac.kr

Hyeyoung Park (Corresponding Author)
School of Electrical Engineering and Computer Science
Kyungpook National University
Daegu, KOREA
hypark@knu.ac.kr

ABSTRACT

Text-guided image retrieval (TGIR) aims to retrieve appropriate target images based on user feedback for a reference image. Existing methods employ global-level representations to model changes in the query by combining global feature vectors from the reference image and feedback text. However, these methods have limitations in capturing local image changes indicated by attribute words in the feedback text, as they do not actively address these local changes during the query combination process. To address this limitation, we propose a novel local-level feature aggregation (LFA) module and training strategy accompanied by a newly defined loss function. In the LFA module, we introduce a set of trainable attribute anchors to aggregate local features of the image and text in the semantic space. These aggregated local features effectively represent local changes in the query and target images from the perspective of multiple attribute anchors. In addition, the LFA module can be easily integrated with existing global-level feature representation modules which play complementary roles in image retrieval. We validate the effectiveness of our proposed method on two benchmark datasets, achieving considerable performance improvement.

Keywords—Text-guided image retrieval, Interactive image retrieval, Local-feature alignment, Multimodal retrieval.

1. Introduction

Text-guided image retrieval (TGIR) task aims to retrieve appropriate target images from a database by reflecting the textual feedback of users for a given reference image. With advances in multimodal representation that integrates vision and language information, several studies have achieved successful results in this field [1,2,3,5,6,10,35]. As a general approach to text-guided image retrieval, the feature vectors of the reference image and feedback text are obtained from image and text feature extraction modules respectively, and they are combined in a composing module to obtain a query representation. The representation of the target image is also obtained using the same image feature extraction module, and its matching score with the query representation is calculated.

The main challenge in this approach is how to combine two different modalities to obtain a query representation. Unlike conventional cross-modal retrieval where two different modality inputs have the same semantic meaning, we need to compose a query representation by combining partial information from two different modalities (i.e., reference image and feedback text).

To tackle the TGIR task, previous studies [1,2] extracted each feature vector defining the reference image and feedback text using a convolutional neural network (CNN) and a long short-term memory (LSTM), respectively. A query vector was then obtained by mapping the two feature vectors onto a common embedding space and combining them by simple operations such as residual connection or concatenation. The representation of the query was matched with the target image feature vector. Also, several studies [12,14] introduced an attention mechanism to obtain refined representations of the reference and target images. In these representations, the feedback text was used as an attention signal to reflect the required changes, and the improved retrieval performance was achieved.

However, these global-level representations could not directly touch the local image features related to the specific attribute words in the feedback text. In addition, expressing the various feedback signals with a single text feature may not sufficiently represent the required changes in image regions. For example, feedback text such as “is short sleeved and has a stripe” should take into account local regions in the image corresponding to the multiple attributes such as length, color, and pattern.

To address this problem, we propose a local-level feature aggregation (LFA) module that represents the query and target features using trainable multiple semantic attribute anchors. In the proposed LFA module, we represent local image and feedback text using the local-level components using attribute anchors. We use multiple trainable anchors to represent various attributes inherent in the query and target data; thus, the final representation is composed of multiple component vectors based on different anchors.

Attribute anchors, which play an essential role in the proposed local representation cannot be pre-defined and need to be discovered

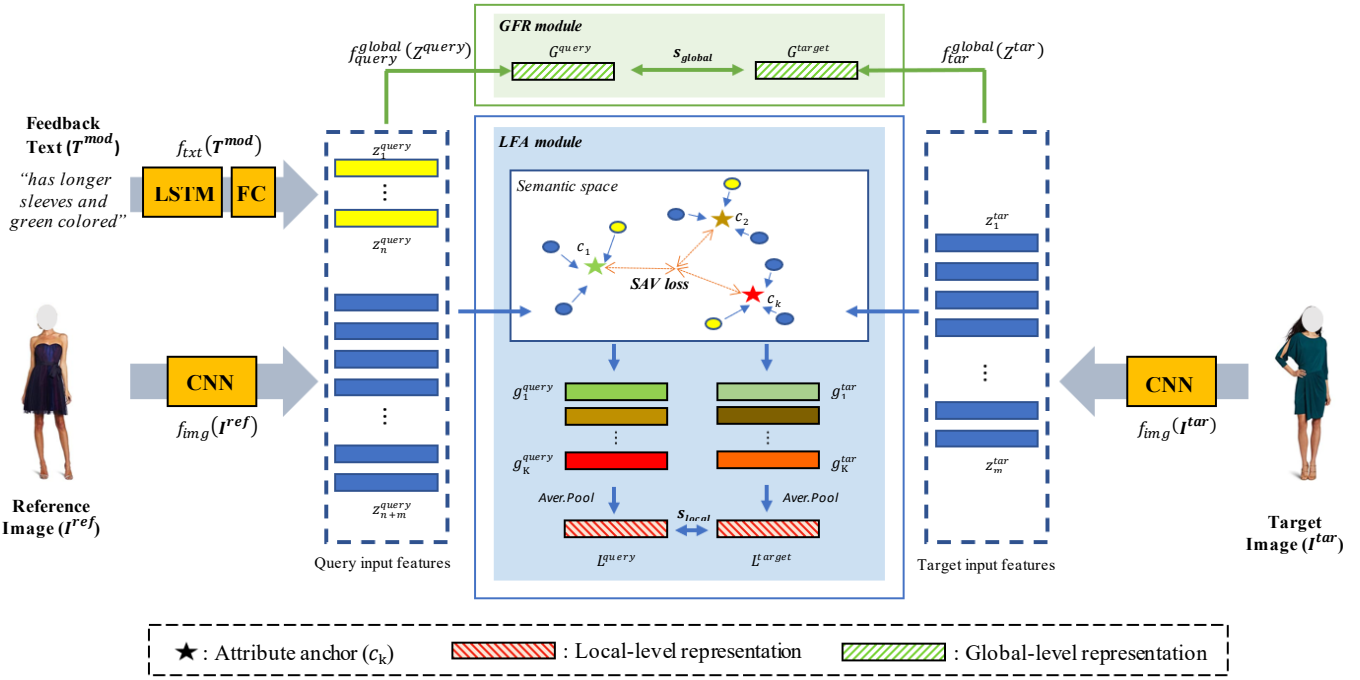

Figure 2. The overall architecture of the proposed model. The text and image (reference/target) features are extracted by respective feature extraction modules. In the blue region (LFA module), query input features of image patches and text words are aggregated around attribute anchors for local-level representation, and target input features are represented as a same manner. In the LFA module, a SAV loss is used to enhance the anchor representations. In the green region (GFR module), query and target input features are integrated into a single vector to produce global-level representation. The final matching score can be obtained as the sum of the similarity scores computed in the LFA and GFR modules.

because the necessary semantic attributes depend on the given retrieval task and dataset. To obtain the discriminative representations of the attribute anchors through learning, we propose a new semantic attribute variance (SAV) loss. The proposed loss prevents the trainable attribute anchors from being biased in some area of the semantic space and allows attribute anchors to have diverse representations.

In addition, to take advantage of existing approaches that employ a global representation of text and image, we incorporate a global-level feature representation (GFR) module that expresses the overall context change for query and target representations. The entire process, including the LFA and GFR modules, is trained in an end-to-end manner, and its notable retrieval performance is verified through the several benchmark datasets. We show that our model gives a significant performance improvement over existing methods and achieves considerable retrieval improvement on two benchmark datasets.

2. Related Works

Text-guided image retrieval is a problem that retrieves appropriate target images from a database when the query is given as a tuple of two components: a reference image and a feedback text requesting its modification. Since a query has two components with different modalities, many studies [1,2,3] have proposed composing modules to combine the reference image and feedback text into a single query representation. The composed query vector can then be used to compute a semantic similarity to the feature vectors of the target image. Therefore, the development of a good composing module is at the core of this approach.

In the early study [1] on the composing module, the TIRG model first extracts text and image feature vectors through LSTM and CNN, respectively. Then, a composing query vector is obtained through an operation between the two features using a gated unit and residual connection. Since then, more sophisticated models [11,13] have been developed that use the hierarchical structure of CNN to compose a query vector. These models represent a query by combining a feedback text feature with image features in each hierarchical layer (high, middle and low) of the CNN feature map, thereby significantly improving retrieval performance. In addition, recent models such as [12,14] have strengthened the feature composition capability by using the feedback feature vector as an attention signal to the image features.

Existing composing modules primarily focus on global-level features when representing queries and targets, and they do not effectively capture changes in specific local-level features of input components. Meanwhile, the proposed model actively uses local-level information by aggregating local features and shows a significant performance improvement.

3. Proposed Methods

3.1 Problem Definition

In this section, we first define a text-guided image retrieval problem. The given data is a set of triplets consisting of two images and a text and is expressed as $\mathcal{D} = \{(I_i^{ref}, T_i^{mod}, I_i^{tar})\}_{i=1...N}$. In the i-th triplet, I_i^{ref} denotes the reference image, T_i^{mod} denotes the feedback text expressing the feedback of a user, and I_i^{tar} denotes the target image to be matched with the query (reference image+feedback text).

The purpose of the retrieval system is to find the appropriate I_i^{tar} through evalutating similarities between candidate images in the database and the given query (I_i^{ref} and T_i^{mod}). As shown in Figure 2, the proposed model obtains a couple of representation pairs, (L^{query}, L^{target}) and (G^{query}, G^{target}), through the two modules, LFA and GFR respectively, when a query (I_i^{ref}, T_i^{mod}) and a target (I_i^{tar}) are given. These representation pairs are used to compute the similarity between query and target to obtain the final score.

3.2 Local-level Feature Extraction

We describe the extraction of input features for the proposed modules, which will be explained in Sections 3.3 and 3.4. The input features for the reference and target images are obtained through the image feature extraction module as follows:

$$\begin{aligned} Z^{ref} &= f_{img}(I^{ref}) = [z_1^{ref}, z_2^{ref}, \dots, z_m^{ref}], \\ Z^{tar} &= f_{img}(I^{tar}) = [z_1^{tar}, z_2^{tar}, \dots, z_m^{tar}], \end{aligned} \quad (1)$$

where f_{img} computes dim planes of size $w \times h$ for the input image by using a pre-trained CNN model, then f_{img} vectorizes each plane, and rearranges them into $m(w \times h)$ local features. Here, $Z^{ref} \in \mathbb{R}^{m \times dim}$, $Z^{tar} \in \mathbb{R}^{m \times dim}$, and m and $dims$ denote the number and dimension of local features, respectively. Note that the i-th triplet sample of the dataset is expressed as $I^{ref}, I^{tar}, T^{mod}$ to simplify the notation in Sections 3.2-3.4.

Similar to the image representation, the input features for the feedback text are obtained through the text feature extraction module as follows:

$$Z^{mod} = f_{txt}(T^{mod}) = [z_1^{mod}, z_2^{mod}, \dots, z_n^{mod}], \quad (2)$$

where f_{txt} uses a bidirectional gated recurrent unit (Bi-GRU) or LSTM and adds a fully connected layer to match the dimensions of features with the image features. Therefore, $Z^{mod} \in \mathbb{R}^{n \times dim}$ and n denotes the number of words in the feedback text. By concatenating Z^{ref} and Z^{mod}, we obtain a representation consisting of local image features and word-level features for the following query input:

$$\begin{aligned} Z^{query} &= [Z^{ref}, Z^{mod}] \\ &= [z_1^{query}, z_2^{query} \dots, z_{(n+m)}^{query}]. \end{aligned} \quad (3)$$

3.3 Local Feature Aggregation using Attribute Anchors

In the TGIR task, a feedback text is composed of several words and each word interacts with a specific local region in the reference image. However, in existing methods [1,15,18] that use only the global-level representation, two query components (i.e., reference image and feedback text) are integrated as a single vector. The specific parts of the image to be changed are influenced by the information of other words in the text. This representation method was not sufficient to model the partial change of the image by several words in a sentence. To solve this problem, we propose a novel local feature aggregation (LFA) module that can well represent changes in local-level input features. Inspired by the recent study [4] that integrates various video-related information using the shared semantic centers, we propose to use trainable attribute anchor vectors to capture the changes of local components around anchors.

In this section, we assume that K attribute anchors $c_k \in \mathbb{R}^{dim}$ $(k = 1, \dots, K)$ are given, and describe how to aggregate local features using them. Since the anchor vectors are trainable, they are optimized with other parameters during the training process described in Section 3.5. As shown in Figure 2, the local-level input features entering the LFA module are aggregated around K attribute anchors in the semantic space. The query and target are re-represented using a weighted sum of the residual vectors (difference vector between each anchor and local features) and the pooling process.

Specifically, when Z^{query} is given as input to the LFA module, the degree of assignment of the l-th query feature z_l^{query} for the j-th attribute anchor in the semantic space can be computed as follows:

$$a_{l,j} = \frac{\exp\,(z_l^{query} w_j^{\intercal})}{\sum_{k=1}^{K} \exp\,(z_l^{query} w_k^{\intercal})}, \quad (4)$$

where w_k denotes a trainable weight vector for the k-th anchor. Using this degree of assignment as a weight, we obtain the representation related to the j-th attribute anchor for the query components Z^{query} as follows:

$$g_j^{query} = normalize\left(\sum_{l=1}^{n+m} a_{l,j}\left(z_l^{query} - c_j\right)\right), \quad (5)$$

where $normalize$ denotes a unit L_2 nomalization. Specifically, the local-level components related to a specific attribute anchor(c_j) are assembled around the anchor with a high degree of assignment($a_{l,j}$) in the semantic space. Through this process, g_j^{query} integrates the local-level query components as a viewpoint of the j-th attribute anchor. Therefore, the property of each anchor can be defined as the aggregated local features and we call it an *attribute*.

Finally, these attribute anchors are merged by average pooling, and the local-level query representation is defined as follows:

$$L^{query} = avepool\{g_1^{query}, g_2^{query}, \dots g_K^{query}\}. \quad (6)$$

The target representation is processed in a similar way to the query. The local features $[z_1^{tar}, \dots, z_m^{tar}]$ of the target image are aggregated around each attribute anchor c_j such as:

$$g_j^{tar} = normalize\left(\sum_{l=1}^{m} \frac{\exp\left(z_l^{tar} w_j^{\intercal}\right)}{\sum_{k=1}^{K} \exp\,(z_l^{tar} w_k^{\intercal})}\left(z_l^{tar} - c_j\right)\right). \quad (7)$$

By applying average pooling, a local-level target representation L^{target} can be obtained. The objective of TGIR task is matching query with target. According to the task definition, L^{query} and L^{target} obtained from a single triplet $(I_i^{ref}, T_i^{mod}, I_i^{tar})$ should

represent the same semantics. This is achieved through training to maximize the similarity between the two vectors:

$$s_{local} = \kappa(L^{query}, L^{target}), \quad (8)$$

where the similarity kernel κ applies the dot-product similarity.

3.4 Global-level Feature Representation

With the local-level feature representation using attribute anchors, the existing global-level representation which captures the overall contextual change can still be used as a complementary perspective role. We introduce a global-level feature representation (GFR) module that can represent global-level properties in the query and target. In the GFR module, the input data is represented as a single vector. In many previous studies [1,2,15], the feature vectors of the reference image and the feedback text are combined by vector operations such as residual connection [1,2] and concatenation [15,28] to obtain a query vector and compare it with the encoded target vector.

Although there are several existing global-level representation methods, we adopted three representative methods in the experiments, the residual connection [1], concatenation [28] and attention-based methods [14] which are popular in the TGIR task. Using these methods, the global-level query representation G^{query} and the target representation G^{target} are obtained as follows:

$$G^{query} = f_{query}^{global}(Z^{ref}, Z^{mod}), \quad (9)$$

$$G^{target} = f_{target}^{global}(Z^{tar}), \quad (10)$$

where f_{query}^{global} and f_{target}^{global} denote a composing module and mapping function that we adopts from the previous works [1,14,28] for the global-level representation. Finally, the similarity between these two global-level representations is computed similar to the LFA module as follows:

$$s_{global} = \kappa(G^{query}, G^{taret}), \quad (11)$$

where κ applies the dot product similarity, as in Section 3.3.

3.5 Training and Inference

In the training stage, the proposed LFA module including attribute anchors and GFR modules are trained simultaneously in an end-to-end manner. Similar to previous research in [1,2], we use the batch-based classification loss, which is widely used for TGIR task, as the objective function for learning local and global-level representations:

$$L_{local} = -\frac{1}{|B|}\sum_{i=1}^{|B|} log \frac{\exp\{\kappa(L_i^{query}, L_i^{query})\}}{\sum_j \exp\{\kappa(L_i^{query}, L_j^{query})\}}, \quad (12)$$

$$L_{global} = -\frac{1}{|B|}\sum_{i=1}^{|B|} log \frac{\exp\{\kappa(G_i^{query}, G_i^{query})\}}{\sum_j \exp\{\kappa(G_i^{query}, G_j^{query})\}}, \quad (13)$$

where B is the mini-batch training set.

In addition, we propose a novel loss that increases the variance of the attribute anchors to diversify the representation of the LFA module. Inspired by the method of improving representation diversity among prototypes in codebook-style video representations [24], the proposed semantic attribute variance (SAV) loss is designed to increase the variance between randomly designated attribute anchors in the semantic space. Therefore, the attribute anchors are prevented from being similar to each other during the learning process. Accordingly, the unbiased attribute anchors can produce diverse representations in the LFA module. This improves retrieval performance, as confirmed via an ablation study. The SAV loss is defined as follows:

$$L_{SAV} = \frac{1}{D}\sum_{d=1}^{D} \max(0, \gamma - \sqrt{Var(c^d) + \epsilon}), \quad (14)$$

where $Var(c^d)$ denotes the variance of the d-th element values in anchor vectors $c_1, \ldots, c_K$, D denotes the dimensionality of anchor vectors, γ is a constant, and ϵ represents a scalar value to prevent errors in numerical calculations. The total loss function for training is defined as follows:

$$L_{total} = \alpha L_{local} + L_{global} + L_{SAV}, \quad (15)$$

where α represents a hypermeter for balanced training during the learning process.

In the inference stage, for the test query (I^{tst}, T^{tst}) given as a pair of images and text, the matching score for the j-th candidate image I_j^{cand} in the database is computed by $s_{local}(I^{tst}, T^{tst}, I_j^{cand}) + s_{global}(I^{tst}, T^{tst}, I_j^{cand})$. For each query, we take top@K images with high similarity scores from all candidate images as retrieval results.

4. Experiments

4.1 Experimental Settings

We verified the performance of the proposed model on two benchmark datasets (FashionIQ and Shoes). For the FashionIQ and Shoes datasets, we used ResNet-50 [26] for the image feature extractor in order to have a fair comparison with previous results. The image feature extractors were pre-trained on ImageNet-1k, and the weights were fixed during the training of the proposed modules. For the text feature extractor, trainable LSTM and Bi-GRU were used. The pre-trained GloVe [19] was used for word embedding in the text preprocessing.

As reported in Section 3.4, the GFR module operates in three ways: GFR_{TIRG} refers to the residual connection-based method [1], $GFR_{Combiner}$ refers to the concatenation-based method [27] and GFR_{ARTEMIS} refers to the attention-based method [14]. In the whole training experiment, we set the batch size to 32, the hidden state of LSTM and Bi-GRU to 1024, and the optimizer as AdamW [20]. Furthermore, α of the loss was set to 0.05 and ϵ of the SAV loss was set to 0.0001. The learning rate has a decay of 0.5 per 10 epochs from the initial value of 0.0005. We use a single NVIDIA RTX 3090 GPU for the experiments, The Recall@K metric was used to evaluate the retrieval performance.

4.2 Experiments on Benchmark Datasets

Results on FashionIQ: FashionIQ [21] is a dataset consisting of three fashion-related categories (dress, shirt, and top), 18k training

Table 1. Experiments on FashionIQ dataset. We mark 1st score in red and 2nd score in blue. The results indicated by † are re-implemented, and the results indicated by * are cited from [14]. For the image feature extraction, we used a Resnet-50 model [26] for all comparison models.

Methods	Feature extractor (Text)	Recall@10				Recall@50			
		Dress	Shirt	Toptee	Mean	Dress	Shirt	Toptee	Mean
JVSM* [2]	*LSTM*	10.70	12.00	13.00	11.90	25.90	27.10	26.90	26.63
ComposeAE* [5]	*BERT*	-	-	-	11.80	-	-	-	29.40
TCIR* [17]	*GRU*	19.33	14.47	19.73	17.84	43.52	35.47	44.56	41.18
VAL* [16]	*LSTM*	22.53	22.38	27.53	24.14	44.00	44.15	51.68	46.61
COSMOS* [17]	*BERT*	21.39	16.90	21.32	19.87	44.45	37.49	46.02	42.65
TIRG† [1]	*LSTM*	23.95	19.38	25.37	22.86	49.33	39.99	51.20	46.78
Combiner† [28]	*LSTM*	24.21	19.62	25.83	23.22	49.40	41.30	52.42	47.71
ARTEMIS† [14]	*LSTM*	25.48	20.76	27.69	24.60	51.44	43.96	53.31	49.52
	Bi-GRU	26.60	22.55	29.35	26.12	51.88	44.38	54.03	50.04
Proposed models	*LSTM*	28.34	22.01	29.55	26.75	53.62	47.23	57.85	52.83
$LFA+GFR_{TIRG[1]}$	*Bi-GRU*	29.13	22.28	29.96	27.08	55.60	46.57	57.60	53.19
$LFA+GFR_{Combiner[28]}$	*LSTM*	28.50	22.27	29.62	26.80	54.05	47.40	57.82	53.09
	Bi-GRU	29.15	22.25	29.79	27.06	55.16	46.11	57.93	53.06
$LFA+GFR_{ARTEMIS[14]}$	*LSTM*	27.79	23.16	29.47	26.77	54.21	46.10	56.45	52.19
	Bi-GRU	29.40	23.85	30.85	27.99	55.28	46.03	56.87	52.64

Table 2. Experiments on Shoes dataset. The results indicated by † are re-implemented, and the results indicated by * are cited from [14].

Methods	Recall@K			
	R@1	R@10	R@50	Mean
TIRG† [1]	15.52	48.65	76.49	46.89
Combiner† [28]	16.01	48.98	76.93	47.31
VAL* [16]	16.49	49.12	73.53	46.38
CoSMo* [17]	17.18	51.52	75.83	48.18
ARTEMIS† [14]	18.72	53.11	79.31	50.38
Proposed models $LFA+GFR_{TIRG[1]}$	17.83	51.62	77.97	49.14
$LFA+GFR_{Combiner[28]}$	18.40	52.31	78.04	49.58
$LFA+GFR_{ARTEMIS[14]}$	19.02	54.57	79.33	50.97

triplets, and 12k test triplets. Table 1 shows the retrieval results in terms of recall@10,50. The proposed model achieves considerable improvement results and shows a remarkable performance improvement in all aspects compared with the existing models. In the case of FashionIQ, which contains relatively long feedback text, the ability of the text feature extractors (LSTM/Bi-GRU) is significantly affected. From the viewpoint of performance, a pre-trained language model on large data, such as BERT [25], can be selected as a boosting strategy. In particular, by comparing the retrieval performances of TIRG [1] and Combiner [27] and ARTEMIS [14], the representative global-level representation models, with that of the proposed *LFA+GFR*, we can see the clear improvement in retrieval performance.

Results on Shoes: Shoes dataset [22] comprises a triplet of a shoe reference image, a feedback text, and a target image. This dataset was extracted from the Attribute Discovery Dataset [23], which consists of 9k training triplets and 1.7k test queries.

Table 3. Ablation studies on FashionIQ and Shoes dataset. We used the ARTEMIS [14] model to represent the global-level feature features.

Methods	Proposed components	**Recall@10**	
		FashionIQ (Mean)	Shoes
Proposed model (RN50/Bi-GRU)	Full	27.99	54.57
w/o Local score	GFR SAV	26.32	52.11
w/o Global score	LFA SAV	26.23	51.68
w/o SAV loss	GFR LFA	27.40	53.30

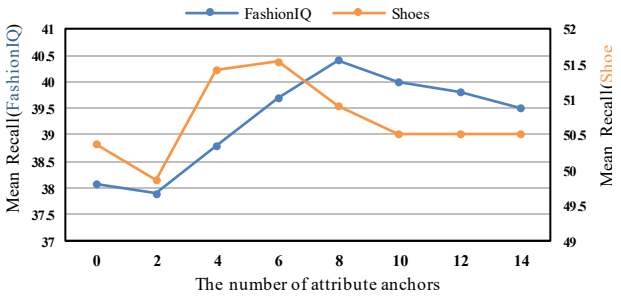

Figure 2. Effect of the number of attribute anchors on retrieval performance.

From Table 2, we can see that the proposed model gives an improved performance over the whole range compared to the existing model. In particular, in the case of Shoes dataset, because the local feature of the image to be changed is relatively simple, the global representation difference between the GFR modules has a greater effect on the retrieval performance than the LFA module. Despite the lack of discriminative local features, the proposed LFA module

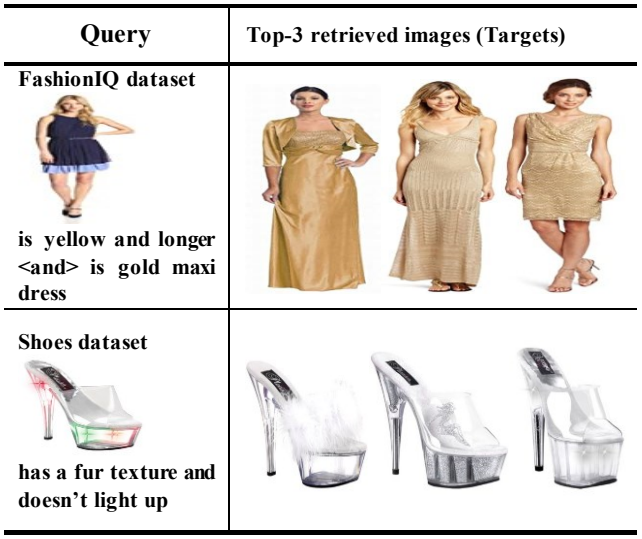

Figure 3. Examples of top-3 retrieved images using the proposed model. Results have higher search rankings from left to right.

successfully improves retrieval performance in all recall metrics when combined with the GFR module.

4.3 Ablation Studies

To more clearly demonstrate the effect of the proposed modules, we performed an ablation study. As can be seen from the results in Table 3, the proposed modules (GFR and LFA) and SAV loss contribute to the improvement of retrieval performance. For the full component analysis of the proposed model, components are removed one by one to verify the effect of the proposed module. It is confirmed that both the local and global-level representation are meaningful and the SAV loss contributes to performance improvement.

We investigated the effect of the number of attribute anchors in the semantic space on retrieval performance. As shown in Figure 2, increasing the number of attribute anchors improves retrieval performance to a certain extent, but beyond this point there is no additional improvement in retrieval performance. For this result, it is speculated that too many anchors lead to semantic redundancy among them. In addition, it is found that the appropriate number of anchors is related to the diversity in the attributes of the given data. Compared to the Shoes dataset, the FashionIQ dataset generally has more diverse attributes, and thus more attribute anchors are required to express the diversity.

4.4 Qualitative Analysis

Figure 3 shows examples of the retrieval results of top-3 by the proposed model. In summary, the retrieved image reflects the two components of the query well. In particular, as shown in FashionIQ dataset, the requested attribute changes may be conflicted in the feedback text (yellow and gold color), and the model suggests various images depending on the requests. Also, as shown in the Shoes dataset, if the attributes in the image to be changed and the feedback texts are simple and clear, the model can easily find the appropriate target images in a database.

5. Conclusion

In this paper, we studied text-guided image retrieval using two representations called the LFA and GFR modules. The LFA module is capable of capturing changes in the local-level features of queries and targets for text-guided image retrieval. In the LFA module, the information from different modalities is well aligned and represented using a representation based on attribute anchors with a new loss function training for the LFA module. We experimentally confirmed the significant improvement in retrieval performance of our model using two benchmark datasets. In the future, the performance of the proposed method can be improved by developing more sophisticated models for input feature extraction as well as for global-level representation modules.

Acknowledgment. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2021-II212068, Artificial Intelligence Innovation Hub).

REFERENCES

[1] Vo, Nam, et al. Composing Text and Image for Image Retrieval - An Empirical Odyssey. In CVPR, 2019.
[2] Chen, Yanbei, and Loris Bazzani. Learning Joint Visual Semantic Matching Embeddings for Language-guided Retrieval. In ECCV, 2020.
[3] Kim, Jongseok, et al. Dual compositional learning in interactive image retrieval. In AAAI, 2021.
[4] Wang, Xiaohan, Linchao Zhu, and Yi Yang. T2vlad: global-local sequence alignment for text-video retrieval. In CVPR, 2021.
[5] Anwaar, Muhammad Umer, Egor Labintcev, and Martin Kleinsteuber. Compositional learning of image-text query for image retrieval. In WACV. 2021.
[6] Wen, Haokun, et al. Comprehensive linguistic-visual composition network for image retrieval. In SIGIR, 2021.
[7] Zhang, Ying, and Huchuan Lu. Deep cross-modal projection learning for image-text matching. In ECCV. 2018.
[8] Li, Junnan, et al. Align before fuse: Vision and language representation learning with momentum distillation. In NIPS, 2021.
[9] Zhang, Feifei, Mingliang Xu, and Changsheng Xu. Geometry sensitive cross-modal reasoning for composed query based image retrieval. In TPAMI, 2021.
[10] Yang, Yuchen, et al. Cross-modal joint prediction and alignment for composed query image retrieval. In ACMMM. 2021.
[11] Gu, Chunbin, et al. Image search with text feedback by deep hierarchical attention mutual information maximization. In ACMMM. 2021.
[12] Jandial, Surgan, et al. SAC: Semantic attention composition for text-conditioned image retrieval. In WACV. 2022.
[13] Zhang, Feifei, et al. Joint attribute manipulation and modality alignment learning for composing text and image to image retrieval. In ACMMM. 2020.
[14] Delmas, Ginger, et al. Artemis: Attention-based retrieval with text-explicit matching and implicit similarity. In ICLR, 2022.
[15] Chawla, Pranit, et al. Leveraging style and content features for text conditioned image retrieval. In CVPR. 2021.
[16] Chen, Yanbei, Shaogang Gong, and Loris Bazzani. Image search with text feedback by visiolinguistic attention learning. In CVPR, 2020.
[17] Lee, Seungmin, Dongwan Kim, and Bohyung Han. Cosmo: Content-style modulation for image retrieval with text feedback. In CVPR, 2021.
[18] Perez, Ethan, et al. Film: Visual reasoning with a general conditioning layer. In AAAI. 2018.
[19] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In EMNLP, 2014.
[20] Loshchilov, Ilya, and Frank Hutter. Decoupled weight decay regularization. In ICLR, 2017.
[21] Wu, Hui, et al. Fashion iq: A new dataset towards retrieving images by natural language feedback. In CVPR, 2021.
[22] Guo, Xiaoxiao, et al. Dialog-based interactive image retrieval. In NIPS, 2018.
[23] Berg, Tamara L., Alexander C. Berg, and Jonathan Shih. Automatic attribute discovery and characterization from noisy web data. In ECCV, 2010.
[24] Lin, Chengzhi, et al. Text-adaptive multiple visual prototype matching for video-text retrieval. In NIPS, 2022.
[25] Devlin, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT, 2019.
[26] Li, Gen, et al. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In AAAI, 2020.
[27] A. Baldrati, et al. Conditioned and composed image retrieval combining and partially fine-tuning CLIP-based features, In CVPRW, 2022.