An Interactive Framework for Image Retrieval

Nikhil Mehta, Fan Yang, Stephen Rawls, Loris Bazzani, Chengwei Su, Emre Barut

Motivation

- Online shopping is extremely popular!
- Customers often go to a search engine, issue queries to find products they want, and the system returns queries
- If the user doesn't like any of the returned products, they must issue a new query

- Instead, the user should be able to issue a modification query!
- Ex in Fig. 1: "I like that skirt, but find it in black"
- This problem is called ***multi-modal product image retrieval***
- ***Task Description:*** *Given a candidate product and a modification query, find the target product*

- Existing Approaches use Deep Neural Networks to tackle this problem
- Several Drawbacks:
- This is a highly connected task -> products are connected via attributes that they share in common, NN's don't take advantage of this
- NN's aren't interpretable, hard to enable **interactivity**
- Our Key Question: How can we solve the multi-modal product image retrieval task better and enable further user interactivity?

Graph Overview

- We propose to use Graph Neural Networks (GNNs)
- Graph Model uses the connectivity between attributes and products to solve the task
- Task is now simpler: Find target products that have similar attributes as the candidate but have the desired modification
- Graphs are also interpretable, we know why products were chosen (what attributes they had) and other products that were considered (based on node embedding similarity)

- Graph structure in Fig. 2 to the right: Products, Attributes, and Modification Queries

- Train Graph for Link Prediction and a Compositional Loss (Anwar et. al)

"I like that skirt, but find it in black"

Fig 1: Example of multi-modal image retrieval. After issuing an initial query, the user is shown the skirt on the left, but wants it black. Thus, they issue a new query with text shown and the image of the skirt. The system finds a similar product with the desired modification, successfully returning the image on the right

Interactions

- Our graph model is very interpretable, so we can support interactions from users
- We explore two forms of interactions:
- Users can provide more information about products, or express their preferences

- **More Metadata:** With this interaction, the user provides the model with more information about a given product
- Ex: "That shirt has long sleeves"
- Product is connected to the attribute with an edge in the graph
- **I Like That!:** With this interaction, the user expresses their content preferences
- User tells the system which groups of products they like in a given context
- Ex: I like these dresses as they all are striped
- System learns from this to predict those dresses together in the future
- Those products are connected in the graph with a special edge type

- Once the interaction is learned (after a brief training round), it can be incorporated into the model with no further training needed!

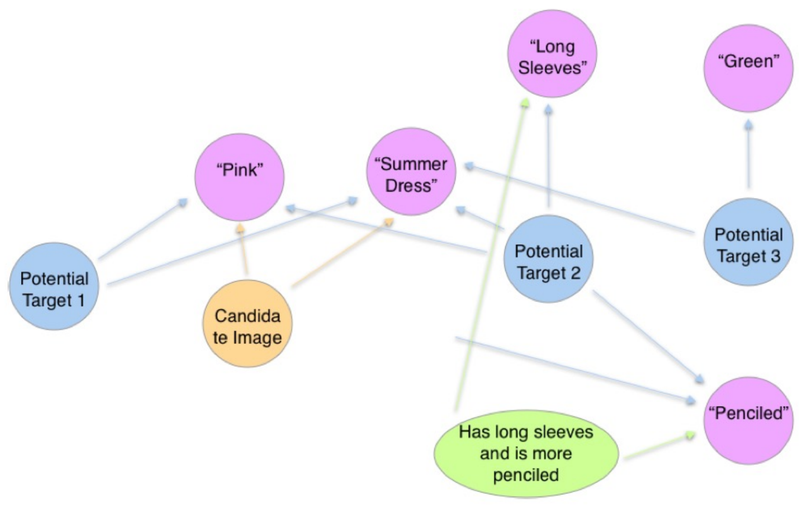

Fig 2: Information Graph capturing interactions between images (candidate: orange background, target: blue background), relative captions (green background), and attributes (pink background)

Datasets

- **Fashion200K**: Dataset that initially consisted of captions from Google Search, was expanded for this task
- Expansion: Choose random candidate images, target images are those with the same caption but with a one word difference
- We get attributes two ways: Either predict them using an off-the-shelf model, or get them from the caption (first word is one attribute, rest of the caption is another)

- **FashionIQ:** Dataset with attributes and images annotated by humans

- **Simulating Interactions**
- For More Metadata, we use the attributes from Fashion200K
- For "I Like That", we randomly select images the user may have liked, using FAISS image similarity search

Results

Model	R@1	R@10	R@50
TIRG Vo et al. (2019)	14.1	42.5	63.8
TIRG with Complete Text Query	14.2	41.9	63.3
TIRG with BERT and Complete Text Query	19.9	51.7	71.8
ComposeAE Anwaar et al. (2021)	22.8	55.3	73.4
Our Graph with Predicted Attributes	16.97	40.0	59.91
Our Graph with Caption Based Attributes	82.1	82.5	87.5

Table 1: Fashion200K results. Our graph with caption based attributes (gotten by splitting the caption into two attributes: the first word and the rest of the caption) achieves performance improvements over baselines.

Model	R@10	R@50	R@100
TIRG with Complete Text Query	3.34	9.18	9.45
TIRG with BERT and Complete Text Query	11.5	28.8	28.9
ComposeAE Anwaar et al. (2021)	11.8	29.4	29.9
Our Graph	6.74	19.09	30.2

Table 2: FashionIQ results. We see improvements over baselines on R@100.

Model	R@1	R@10	R@50
AA1: Graph with Predicted Attributes	16.97	40.0	59.91
AA2: Graph w/ Additional 5 Attributes Tagged per image	15.81	75.77	83.63
AA3: Graph w/ Additional 15 Attributes Tagged per image	22.75	75.63	84.06
AA4: Graph with 1/2 Images With 15 Additional Attributes	8.43	59.71	83.95
AA5: Graph with Additional All Caption Based Attributes	82.1	82.5	87.5

Table 3: Fashion200K More Metadata. We achieve the strongest performance when all images are tagged with all aattributes (AA5), but performance is still strong when using only 5/15 (AA2, AA3), tagging only half the images (AA4), or predicting attributes from an external model (AA1).

Model	R@1	R@10	R@50	# Edges	Conn. Acc.
BB1: Graph with Predicted Attributes	16.97	40.00	59.91	-	-
BB2: I Like That! 1000 Examples	16.55	62.26	77.14	58,400	65.73%

Table 4: Fashion200K I Like Like That Interaction. We can see that the "I Like That" interaction improves performance (BB2 vs BB1), particularly in R@10 and R@50. The model also has high connection accuracy, whis is calculated based on how often it is successful at incorporating the interaction (every time it predicts an image the user liked, it should also predict another one the user liked in its top 10).