Fashion-SegCBIR

Ngày 13 tháng 11 năm 2025

Nội dung trình bày

Giới thiệu bài toán

Giới thiệu bài toán

- **Bối cảnh:** Trong lĩnh vực thời trang, nhu cầu tìm kiếm các sản phẩm có hình ảnh hoặc phong cách tương tự ngày càng phổ biến — ví dụ: tìm chiếc áo, túi xách hoặc phụ kiện giống với ảnh hoặc mô tả đầu vào.
- **Mô tả bài toán:** Cho một đầu vào là **ảnh** hoặc **mô tả bằng văn bản**, hệ thống cần truy xuất các hình ảnh thời trang có nội dung tương ứng hoặc tương tự nhất trong cơ sở dữ liệu.
- **Đặc điểm chính:** Ảnh đầu vào có thể được phân tích để **phát hiện các thành phần thời trang** (như quần áo, giày dép, phụ kiện), từ đó mã hóa có thể tùy chọn tìm kiếm các ảnh tương tự cho từng phụ kiện.
- **Mục tiêu:** Xây dựng một hệ thống **truy xuất ảnh thời trang thông minh**, có khả năng hiểu được mối liên hệ giữa hình ảnh và văn bản, giúp người dùng tìm kiếm sản phẩm tương tự một cách hiệu quả.

Phân đoạn quần áo từ ảnh

1.Mục tiêu

Chức năng phân đoạn quần áo nhận ảnh người dùng tải lên và sinh mặt nạ (mask) phân lớp cho từng bộ phận trang phục hoặc cơ thể. Kết quả mặt nạ hỗ trợ các tác vụ như thay trang phục, tách nền, gợi ý phối đồ và xử lý hậu kỳ theo vùng.

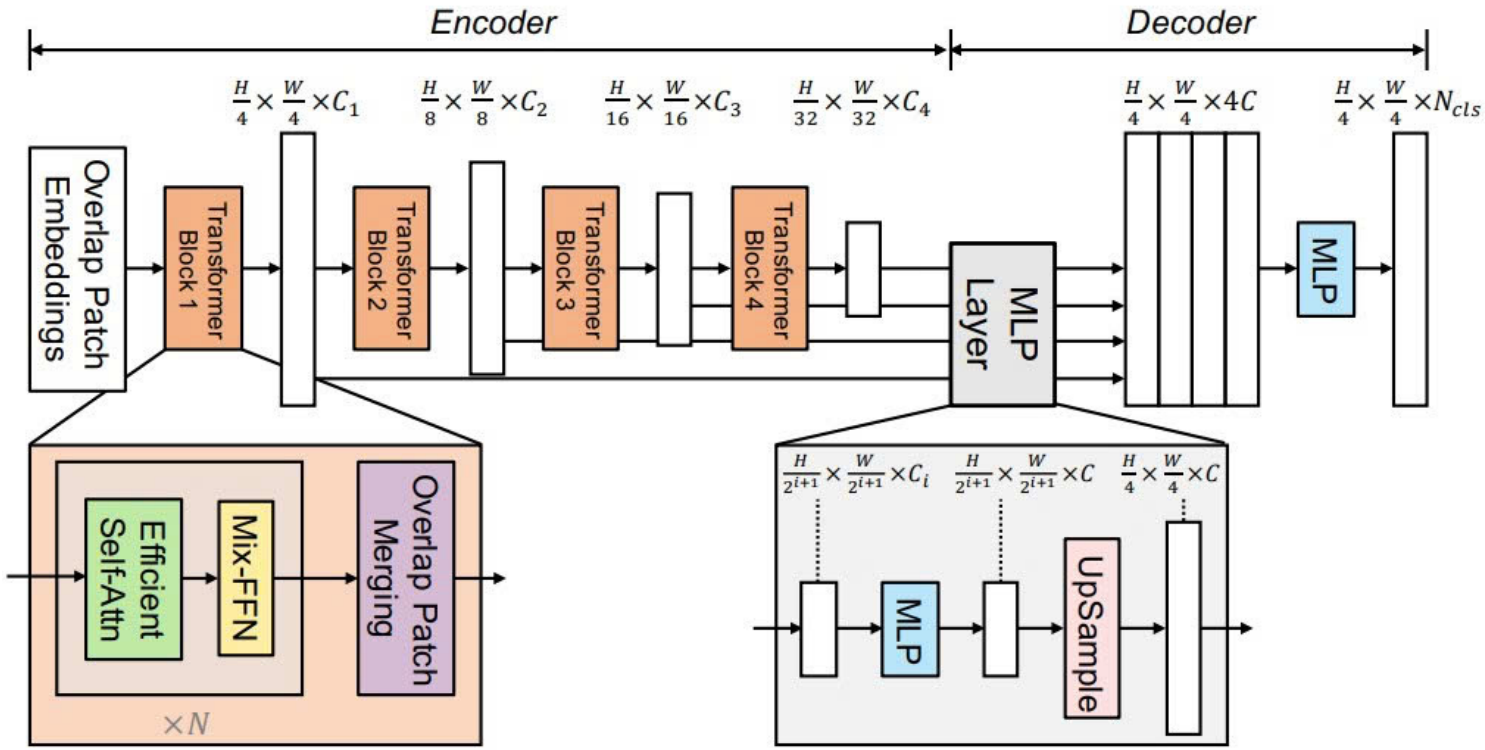
2. Mô hình và dữ liệu

Backbone: SegFormer. Mô hình được khởi tạo từ SegFormer và *fine-tune* cho bài toán **human parsing** (phân đoạn người và trang phục).

Dữ liệu: sử dụng bộ *ATR / Deep Human Parsing* (bản phát hành trên HuggingFace: `mattmdjaga/human_parsing_dataset`), gồm 17 706 cặp ảnh–mask với 18 nhãn:

0: Background 1: Hat 2: Hair 3: Sunglasses
4: Upper-clothes 5: Skirt 6: Pants 7: Dress
8: Belt 9: Left-shoe 10: Right-shoe 11: Face
12: Left-leg 13: Right-leg 14: Left-arm 15: Right-arm
16: Bag 17: Scarf

3.Tóm lược kỹ thuật Segformer



Hình: Sơ đồ tổng quát mô hình Segformer.

3.Tóm lược kỹ thuật SegFormer

SegFormer là khung phân đoạn ngữ nghĩa dựa trên Transformer với hai thành phần chính:

(i) **Encoder Mix Transformer (MiT)** phân cấp;

(ii) **Decoder All-MLP** gọn nhẹ.

Các đặc điểm then chốt:

1. Encoder phân cấp, không dùng positional encoding

- **Overlapping patch embedding:** Thay vì chia patch rời rạc, SegFormer (MiT) sử dụng cửa sổ *chồng lấn* để duy trì tính liên tục cục bộ, tạo đặc trưng đa tỉ lệ ở các mức $\{1/4, 1/8, 1/16, 1/32\}$ so với ảnh gốc.
- **Sequence reduction:** Trong *self-attention*, độ dài chuỗi được rút gọn theo từng *stage* (tỉ lệ $R = [64, 16, 4, 1]$), giúp giảm chi phí tính toán xấp xỉ $O(N^2/R)$ và tăng tốc suy luận trên ảnh có độ phân giải cao.
- **Mix-FFN:** Thay vì dùng positional encoding, SegFormer chèn tích chập sâu 3×3 để đưa tín hiệu vị trí theo cách do dữ liệu dẫn dắt, giúp mô hình ổn định khi suy luận ở độ phân giải khác với lúc huấn luyện.

3.Tóm lược kỹ thuật SegFormer

2. Decoder All-MLP tối giản, hiệu quả:

- Chuẩn hoá số kênh ở mỗi mức đặc trưng bằng MLP, nội suy về cùng tỉ lệ 1/4, ghép (*concat*) và dùng MLP hợp nhất để dự đoán *logits* kích thước $H/4 \times W/4 \times |C|$.
- Do encoder đã bao quát ngữ cảnh xa (ERF lớn), decoder không cần mô-đun nặng (ASPP, PPModule, ...) mà vẫn duy trì chất lượng cao với độ trễ thấp.

3.Tóm lược kỹ thuật SegFormer

3. Ổn định đa độ phân giải, cân bằng độ chính xác–tốc độ–tham số:

- Thiết kế không positional encoding giúp mIoU ít suy giảm khi thay đổi kích thước ảnh đầu vào.
- Các phiên bản B0–B5 mở rộng quy mô encoder, trong khi decoder chỉ chiếm tỉ lệ tham số nhỏ, thuận lợi cho triển khai thời gian thực.

4. So với ViT/SETR:

- SegFormer sinh *đa tỉ lệ* ngay từ encoder; trong khi SETR/ViT thường chỉ cho một mức đặc trưng và cần decoder CNN phức tạp để khôi phục không gian.
- SegFormer đạt cân bằng tốt giữa mIoU và tốc độ với pretrained ImageNet-1K, không đòi hỏi huấn luyện cực lớn.

Kết quả đánh giá sau khi tinh chỉnh mô hình

Sau khi **tinh chỉnh mô hình SegFormer** trên bộ dữ liệu quần áo, kết quả trên tập *test* đạt được như sau:

Chỉ số	**Evaluation Loss**	**Mean Accuracy**	**Mean IoU**
Giá trị	0.15	0.80	0.69

Diễn giải ngắn:

- **Mean Accuracy:** phản ánh tỉ lệ dự đoán đúng trung bình theo từng lớp.
- **Mean IoU:** đo mức chồng lấp giữa vùng dự đoán và nhãn thật trên 18 lớp, đặc biệt nhạy với biên và các vùng trang phục mảnh (ví dụ: *belt, scarf*).

Multimodal model

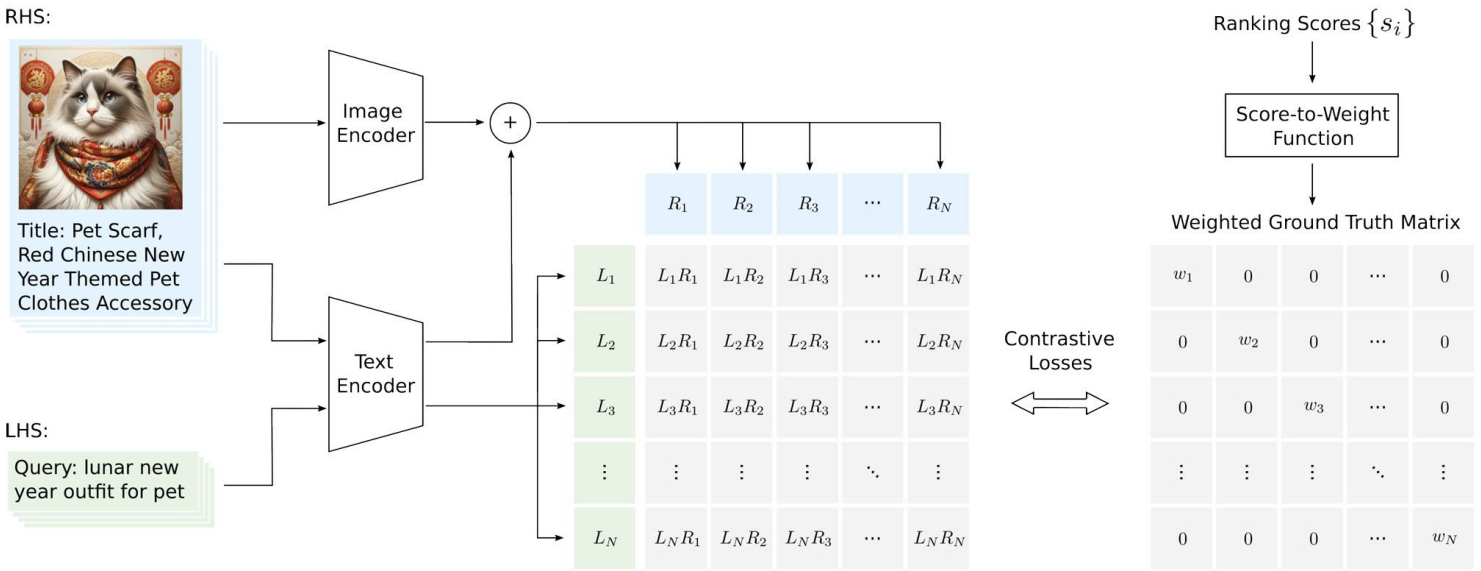
1.Vấn đề với Contrastive learning truyền thống

- Hạn chế cốt lõi: Các mô hình như CLIP chỉ học từ quan hệ nhị phân (tích cực / tiêu cực).
- Vấn đề: Chúng không thể học được thứ tự xếp hạng; mô hình chỉ biết một tài liệu là "liên quan", chứ không biết tài liệu nào "liên quan nhất".
- Hệ quả: Các hệ thống thực tế phải sử dụng kiến trúc hai giai đoạn (retrieval + re-ranking), dẫn đến tăng độ phức tạp, thời gian suy luận và chi phí bảo trì.

2.Generalized Contrastive Learning

- Định nghĩa: GCL là một mô hình huấn luyện được thiết kế để học trực tiếp từ điểm xếp hạng liên tục.
- Mục tiêu: Mã hóa cả thông tin liên quan (relevance) và xếp hạng (ranking) vào một không gian embedding thống nhất.
- Lợi ích: Cho phép xây dựng một hệ thống tìm kiếm và xếp hạng đơn giai đoạn (single-stage) thay vì two-stage, hiệu quả và nhanh chóng.

2.Generalized Contrastive Learning

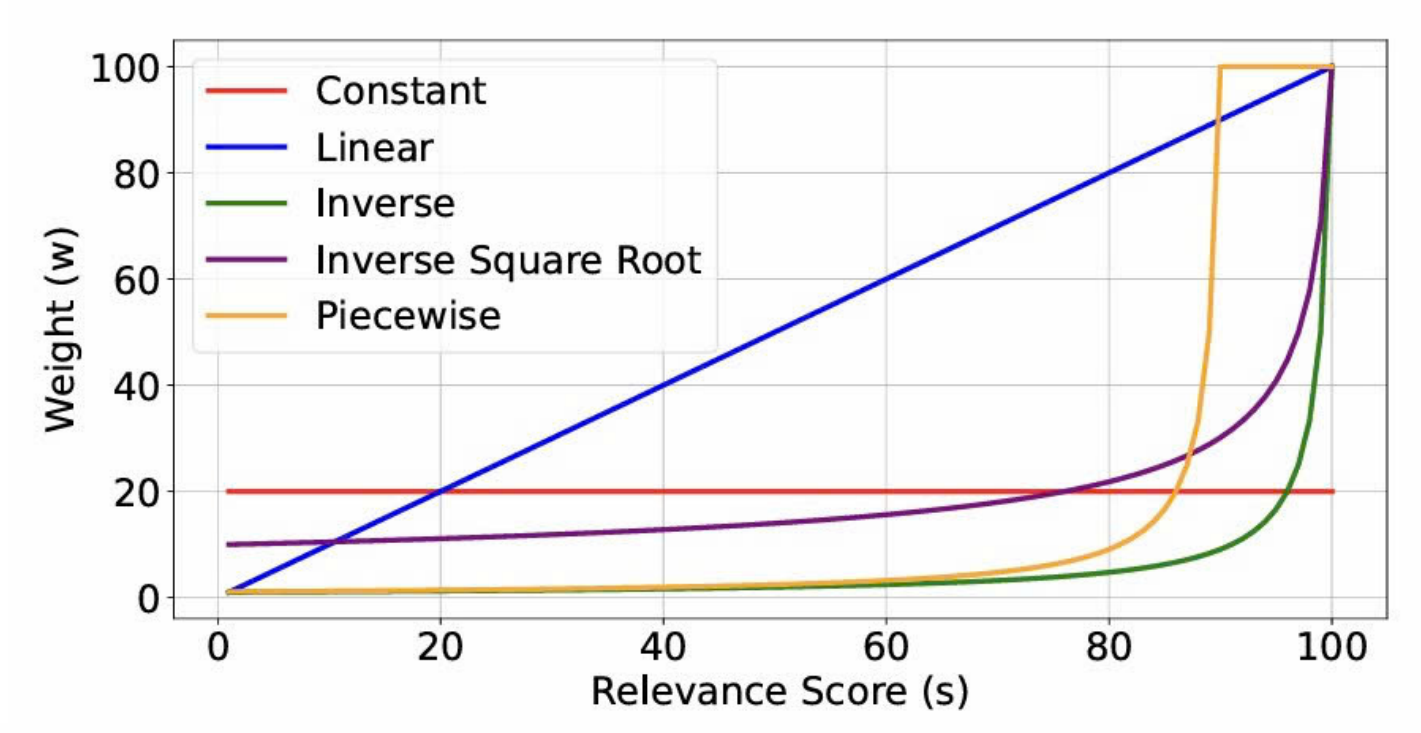


Hình: Kiến trúc tổng quát của mô hình GCL.

3.Cơ chế cốt lõi

- CL truyền thống: Sử dụng các cặp (Pairs): (query, document).
- GCL: Giới thiệu các bộ ba (Triplets): (query, document, weight).
- Trọng số w:
  - Giá trị w được tính toán từ điểm xếp hạng (ranking score s) gốc của cặp (query, document).
  - Việc chuyển đổi được thực hiện thông qua hàm Score-to-Weight Function (STW).
  - Các tài liệu có điểm xếp hạng cao sẽ có trọng số w lớn nhất.

3.Cơ chế cốt lõi (Tiếp)



Hình: Các hàm STW được thử nghiệm.

3.Cơ chế cốt lõi (Tiếp)

- Hàm mất mát (Loss Function): GCL sử dụng *Weighted Cross-Entropy Loss*.
- Nguyên lý hoạt động:
  - Hàm loss này áp dụng hình phạt lớn hơn đối với các dự đoán sai trên các cặp (query, document) có trọng số w cao.

3.Cơ chế cốt lõi (Tiếp)

Weighted cross-entropy loss được định nghĩa là:

$$\mathcal{L}_{WCE}(Z, w) = -\frac{1}{2N}\sum_{i=1}^{N} w_i \left(\log\left(\frac{\exp(Z[i,i])}{\sum_{j=1}^{N}\exp(Z[i,j])}\right) + \log\left(\frac{\exp(Z[i,i])}{\sum_{j=1}^{N}\exp(Z[j,i])}\right)\right)$$

- Giải thích tham số:
  - N: Kích thước batch (số lượng mẫu trong một batch).
  - w_i: Trọng số của cặp (query, document) thứ i, được tính toán từ điểm xếp hạng (ranking score) s_i.
  - Z: Ma trận $N \times N$ chứa điểm tương đồng (tích vô hướng) giữa mọi query và document trong batch.
  - $Z[i,i]$: Điểm tương đồng của cặp "khớp"(positive) thứ i (nằm trên đường chéo chính).
  - $Z[i,j]$: Điểm tương đồng (tích vô hướng) giữa query i và document j.

4.Multi-field Generalized Contrastive Learning

- Vấn đề thực tế: Tài liệu (ví dụ: sản phẩm) thường có nhiều trường như tiêu đề, hình ảnh, mô tả.
- Giải pháp GCL: Hỗ trợ multi-field cho cả query (LHS) và document (RHS).
- Kiến trúc:
  - Mã hóa từng trường (title, image, ...) thành các embedding riêng.
  - Tính toán embedding trung bình có trọng số (weighted average embedding) để đại diện cho toàn bộ tài liệu.
- Điểm quan trọng khắc phục hiệu suất khi chỉ tìm kiếm sử dụng 1 trường:

$$\text{Loss tổng thể} = \text{Loss trên embedding trung bình} + \sum \text{Loss của các cặp trường riêng lẻ}$$

- Lợi ích: Mô hình vẫn hoạt động hiệu quả khi người dùng chỉ tìm kiếm bằng một trường duy nhất (ví dụ: chỉ text, chỉ image).

4.Multi-field Generalized Contrastive Learning

- Công thức hàm mất mát (Loss Function):

$$\mathcal{L} = \mathcal{L}_{WCE}(Z_{avg}, w) + \sum_{j=1}^{m}\sum_{k=1}^{n} \mathcal{L}_{WCE}(Z_{jk}^{LR}, w)$$

- Giải thích tham số:
  - $\mathcal{L}$: Giá trị loss tổng thể (kết hợp).
  - m, n: Lần lượt là số lượng trường (fields) của LHS (query) và RHS (document).
  - w: Vector trọng số (từ ranking score) cho các cặp trong batch.
  - Z_{avg}: Ma trận $(N \times N)$ chứa điểm tương đồng (dot product) của các embedding "trung bình có trọng số" $\hat{L}_{avg}^{f}$ và $\hat{R}_{avg}^{f}$.
  - Z_{jk}^{LR}: Ma trận $(N \times N)$ chứa điểm tương đồng (dot product) giữa trường j của LHS và trường k của RHS.
  - $\mathcal{L}_{WCE}$: Hàm weighted cross-entropy loss (áp dụng cho từng thành phần).

5.Kết quả thực nghiệm

Average evaluation results on 6 public multimodal fashion datasets: Atlas, DeepFashion (In-shop), DeepFashion (Multimodal), Fashion200k, KAGL, and Polyvore.

Model	**AvgRecall**	**Recall@1**	**Recall@10**	**MRR**
Marqo-FashionSigLIP	0.231	0.121	0.340	0.239
FashionCLIP2.0	0.163	0.077	0.249	0.165
OpenFashionCLIP	0.132	0.060	0.204	0.135
ViT-B-16-laion2b_s34b_b88k	0.174	0.088	0.261	0.180
ViT-B-16-SigLIP-webli	0.212	0.111	0.314	0.214

Bảng: Average Text-To-Image Retrieval Performance (6 Datasets)

5.Kết quả thực nghiệm

Average evaluation results on 5 public fashion datasets.

Model	**AvgP**	**P@1**	**P@10**	**MRR**
Marqo-FashionSigLIP	0.737	0.758	0.716	0.812
FashionCLIP2.0	0.684	0.681	0.686	0.741
OpenFashionCLIP	0.646	0.653	0.639	0.720
ViT-B-16-laion2b_s34b_b88k	0.662	0.673	0.652	0.743
ViT-B-16-SigLIP-webli	0.688	0.690	0.685	0.751

Bảng: Average Category-To-Product Retrieval Performance (5 Datasets)

5.Kết quả thực nghiệm

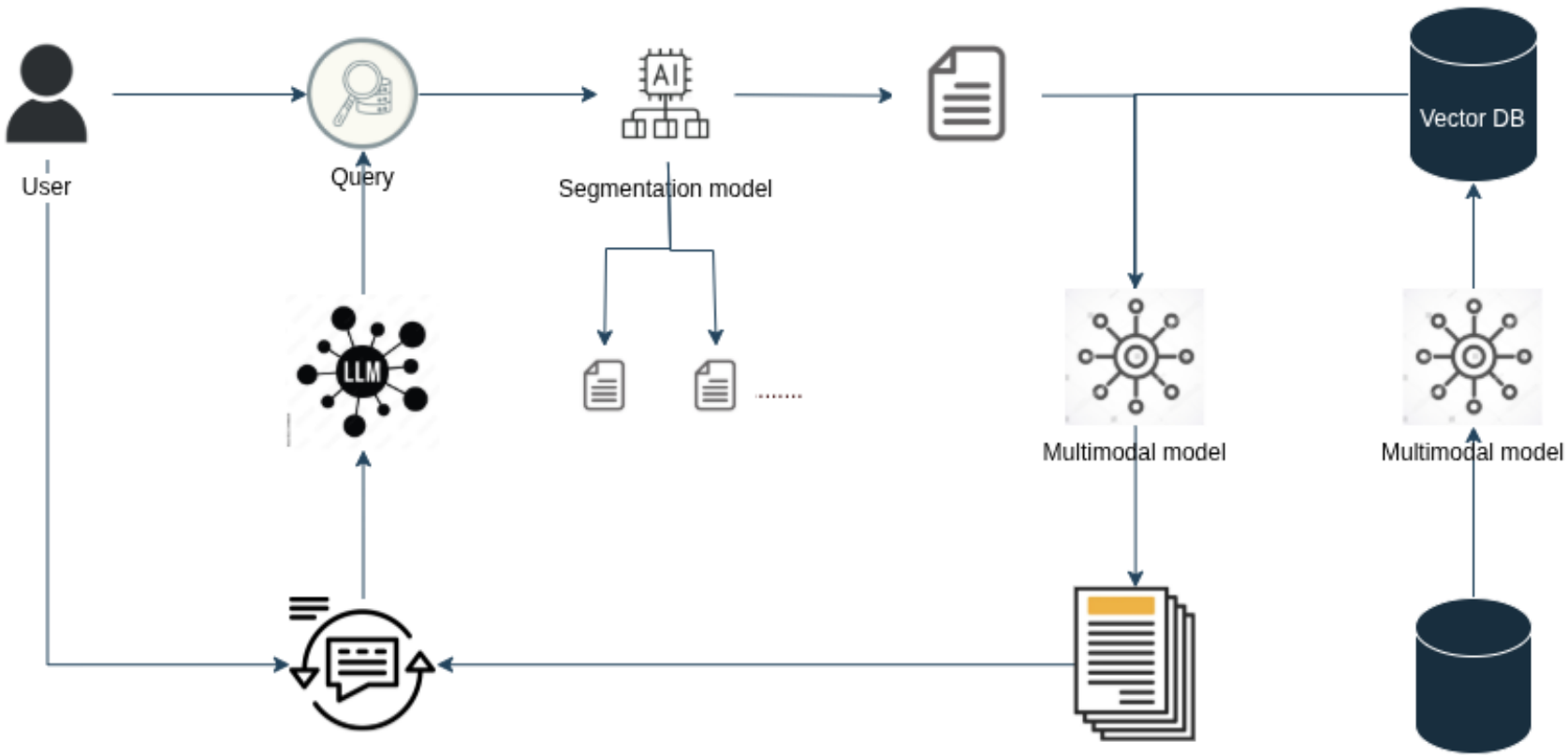
Average evaluation results on 4 public fashion datasets.

Model	**AvgP**	**P@1**	**P@10**	**MRR**
Marqo-FashionSigLIP	0.725	0.767	0.683	0.811
FashionCLIP2.0	0.657	0.676	0.638	0.733
OpenFashionCLIP	0.598	0.619	0.578	0.689
ViT-B-16-laion2b_s34b_b88k	0.638	0.651	0.624	0.712
ViT-B-16-SigLIP-webli	0.643	0.643	0.643	0.726

Bảng: Average Sub-Category-To-Product Retrieval Performance (4 Datasets)

Pipeline

Pipeline Overview



Hình minh hoạ quy trình tổng quát.

Pipeline Components

- **Dataset:** `ceyda/fashion-products-small`
  - Bộ dữ liệu sản phẩm thời trang (hình ảnh, mô tả, danh mục).
- **Segmentation Model:** `mattmdjaga/segformer_b2_clothes`
  - Mô hình SegFormer B2 dùng để tách vùng quần áo trên ảnh.
- **Multi-modal Model:** `Marqo/marqo-fashionSigLIP`
  - Mô hình học đa phương thức (text + image) cho truy vấn sản phẩm thời trang.
- **LLM:** `Gemini 2.5 Flash`
  - Mô hình ngôn ngữ lớn hỗ trợ phân tích và cải thiện truy vấn từ phản hồi của người dùng.

Các chỉ số đánh giá (Metrics)

- **1. Precision@k**

$$\text{Precision@k} = \frac{\text{Số lượng kết quả đúng trong top-}k}{k}$$

Đo lường mức độ chính xác của các kết quả được mô hình trả về trong top-k. $\rightarrow$ *Càng cao, mô hình càng ít trả về kết quả sai.*

- **2. Recall@k**

$$\text{Recall@k} = \frac{\text{Số lượng kết quả đúng trong top-}k}{\text{Tổng số kết quả đúng trong toàn bộ tập dữ liệu}}$$

Thể hiện khả năng của mô hình trong việc tìm được các kết quả đúng. $\rightarrow$ *Càng cao, mô hình bao phủ càng tốt.*

Ví dụ: Nếu trong top-5 kết quả có 3 kết quả đúng và tổng cộng có 6 kết quả đúng trong toàn bộ dữ liệu:

$$\textit{Precision}@5 = \frac{3}{5} = 0.6, \quad \textit{Recall}@5 = \frac{3}{6} = 0.5$$

Chỉ số đánh giá: nDCG@k (Normalized Discounted Cumulative Gain)

- **Công thức:**

$$\text{nDCG@k} = \frac{DCG@k}{IDCG@k},$$

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)}, \quad IDCG@k = \sum_{i=1}^{k} \frac{rel_i^*}{\log_2(i+1)}$$

- **Giải thích các ký hiệu:**
  - rel_i: mức độ liên quan của kết quả tại vị trí i theo ground truth. 1 nếu đúng, 0 nếu sai
  - rel_i^*: mức độ liên quan trong xếp hạng lý tưởng (IDCG), tức tất cả kết quả đúng được đặt ở đầu
  - Mô hình chỉ quyết định thứ tự xuất hiện trong DCG
- **Ý nghĩa:** nDCG@k phản ánh chất lượng xếp hạng, ưu tiên kết quả đúng xuất hiện ở vị trí đầu, cho phép so sánh với xếp hạng lý tưởng.

Ví dụ minh họa: tính nDCG@5

- **Giả sử:**
  - Ground truth = {A, C, E}
  - Mô hình dự đoán top-5 = [B, A, E, D, C]
- **Bảng mức độ liên quan rel_i:**

i	$PredictedResult$	$InGroundTruth?$	rel_i
1	B		0
2	A		1
3	E		1
4	D		0
5	C		1

- **Tính DCG@5:**

$$DCG@5 = \frac{0}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} + \frac{0}{\log_2 5} + \frac{1}{\log_2 6} \approx 1.518$$

- **Tính IDCG@5 (xếp lý tưởng):**

$$rel_i^* = [1, 1, 1, 0, 0], \quad IDCG@5 = 1 + 0.631 + 0.5 \approx 2.131$$

Ví dụ minh họa: tính nDCG@5

- **Kết quả nDCG@5:**

$$nDCG@5 = \frac{DCG@5}{IDCG@5} \approx \frac{1.518}{2.131} \approx 0.712$$

- **Diễn giải:** Mô hình đạt khoảng $nDCG@5 \approx 0.712$ (tương đương 71.2%) so với xếp hạng lý tưởng.

Chỉ số đánh giá: mAP (Mean Average Precision)

- **Công thức tổng quát:**

$$\text{mAP} = \frac{1}{Q} \sum_{q=1}^{Q} AP(q)$$

$$AP(q) = \frac{1}{|R_q|} \sum_{k=1}^{n} P(k) \cdot rel_k$$

- **Giải thích các ký hiệu:**
  - Q : tổng số truy vấn (hoặc người dùng)
  - R_q : tập kết quả đúng (ground truth) của truy vấn q
  - $P(k)$: độ chính xác (Precision) tại vị trí k
  - $rel_k = 1$ nếu kết quả tại vị trí k nằm trong ground truth, ngược lại 0
- **Ý nghĩa:** mAP đo độ chính xác trung bình của mô hình trên nhiều truy vấn, phản ánh hiệu quả tổng thể của hệ thống truy xuất thông tin.

Ví dụ minh họa: tính mAP

- Giả sử ta có 2 truy vấn ($Q = 2$):
  - **Query 1:** Ground truth = {A, C, E}, Predicted = [B, A, E, D, C]
  - **Query 2:** Ground truth = {B, D}, Predicted = [B, C, D, E, F]
- **Tính AP cho mỗi truy vấn:**

$$AP(\text{Query 1}) = \frac{1}{3}(P@2 + P@3 + P@5) = \frac{1}{3}\left(\frac{1}{2} + \frac{2}{3} + \frac{3}{5}\right) \approx 0.611$$

$$AP(\text{Query 2}) = \frac{1}{2}(P@1 + P@3) = \frac{1}{2}\left(\frac{1}{1} + \frac{2}{3}\right) \approx 0.833$$

- **Tính mAP:**

$$mAP = \frac{AP(\text{Query 1}) + AP(\text{Query 2})}{2} \approx \frac{0.611 + 0.833}{2} \approx 0.722\ (72.2\%)$$

- **Diễn giải:** Mô hình đạt khoảng 72% độ chính xác trung bình so với ground truth trên 2 truy vấn.

Kết quả thực nghiệm của mô hình

Metric	Giá trị
Precision@1	0.42
Recall@1	0.42
nDCG@1	0.42
Precision@5	0.45
Recall@5	0.60
nDCG@5	0.52
mAP	0.50

Bảng: Kết quả thực nghiệm của mô hình.

Thank You!