MAPS: Multimodal Attention for Product Similarity

Nilotpal Das[*], Aniket Joshi[*], Promod Yenigalla, Gourav Agrwal
Retail Business Services
Amazon
{nilodas, anikjosh, promy, agrwag}@amazon.com

Abstract

Learning to identify similar products in the e-commerce domain has widespread applications such as ensuring consistent grouping of the products in the catalog, avoiding duplicates in the search results, etc. Here, we address the problem of learning product similarity for highly challenging real-world data from the Amazon catalog. We define it as a metric learning problem, where similar products are projected close to each other and dissimilar ones are projected further apart. To this end, we propose a scalable end-to-end multimodal framework for product representation learning in a weakly supervised setting using raw data from the catalog. This includes product images as well as textual attributes like product title and category information. The model uses the image as the primary source of information, while the title helps the model focus on relevant regions in the image by ignoring the background clutter. To validate our approach, we created multimodal datasets covering three broad product categories, where we achieve up to 10% improvement in precision compared to state-of-the-art multimodal benchmark. Along with this, we also incorporate several effective heuristics for training data generation, which further complements the overall training. Additionally, we demonstrate that incorporating the product title makes the model scale effectively across multiple product categories.

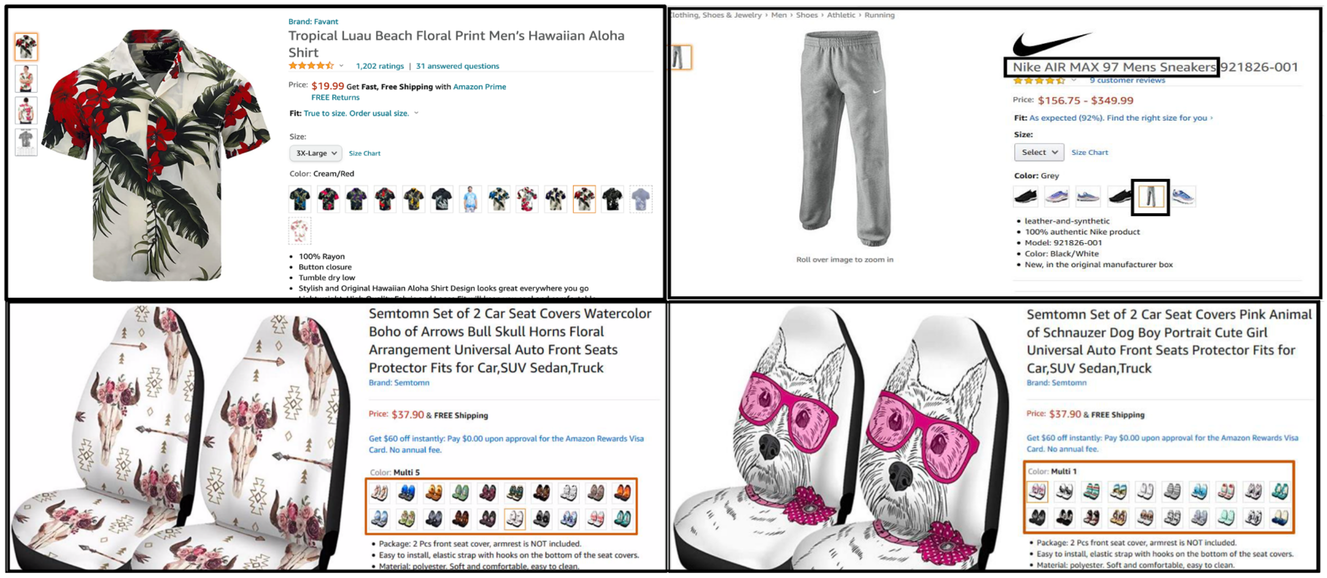
1. Introduction

E-commerce catalog houses not only a rich selection of products but also a great deal of variations within a particular type of product. In order to allow customers make an informed choice specific to their needs, in many e-commerce catalog similar items are grouped together as a variation in a single detail page. These items are the same primary base product but vary across specific themes. Examples of such themes are size and color for shoes, volume and fragrance for detergents and capacity for hard drives. Taking an example of a t-shirt that is exactly the same in every way, except it comes in 8 colours and 5 sizes, it results in 40 different items. Variation listings (a.k.a. variations) allow to group and display different varieties of the same t-shirt on a single detail page which lead to a good shopping experience. Many e-commerce sites configure variations as a parent-child structure. The parent-item is a non-buyable entity which identifies the whole variation listing. In this regard, the parent-item serves as a proxy for the base product identifier. The child-items represent the actual physical items that can be purchased. For our discussion, we will use the term item and child-item interchangeably.

While variations have lots of benefits, creating incorrect variations leads to customer confusion and frustration. The two major catalog quality defects associated with variations are 1) Duplicate Variation (DV) and 2) Inconsistent Variation (IV). Ideally, a variation listing should contain a single base product with all of the different available options i.e, w.r.t. color, style, size, etc., displayed in a single detail page. In DV, items of the same base product are split across multiple variation listings, resulting in customers unable to explore all the product options in a single detail page. It also causes duplicate products appearing in the search results, which is not a desirable customer experience. A variation listing is inconsistent when more than one type of product is grouped together within a single variation. A few examples are displayed in figure 1. Such cases create a confusing customer experience as customers cannot be sure of the identity of the product they are buying. Further, it can lead to bad buying decisions as the product content, and customer reviews get shared between unrelated products. It is therefore important to ensure consistent and comprehensive groupings of items on the detail page.

It is clear that the fundamental requirement in order to detect variation defects is to be able to compare different items and identify whether they are the same or different products. This problem proves to be a challenging task primarily due to a large number of products, their high heterogeneity, lack of labeled data and varying levels of data quality. One simple solution to this is to leverage the universal product identifiers

[*]equal contribution.



Brand: Favant

Tropical Luau Beach Floral Print Men's Hawaiian Aloha Shirt

1,202 ratings | 31 answered questions

Price: $19.99 Get **Fast, Free Shipping** with Amazon Prime

FREE Returns

Fit: True to size. Order usual size.

Size: 3X-Large Size Chart

Color: Cream/Red

- 100% Rayon
- Button closure
- Tumble dry low
- Stylish and Original Hawaiian Aloha Shirt Design looks great everywhere you go

lothing, Shoes & Jewelry › Men › Shoes › Athletic › Running

Nike AIR MAX 97 Mens Sneakers 921826-001

9 customer reviews

Price: $156.75 - $349.99

Fit: As expected (92%). Find the right size for you ›

Size: Select Size Chart

Color: Grey

- leather-and-synthetic
- 100% authentic Nike product
- Model: 921826-001
- Color: Black/White
- New, in the original manufacturer box

Roll over image to zoom in

Semtomn Set of 2 Car Seat Covers Watercolor Boho of Arrows Bull Skull Horns Floral Arrangement Universal Auto Front Seats Protector Fits for Car,SUV Sedan,Truck

Brand: Semtomn

Price: $37.90 & **FREE Shipping**

Get $60 off instantly: Pay $0.00 upon approval for the Amazon Rewards Visa Card. No annual fee.

Color: **Multi 5**

- Package: 2 Pcs front seat cover, armrest is NOT included.
- Easy to install, elastic strap with hooks on the bottom of the seat covers.
- Material: polyester. Soft and comfortable, easy to clean.

Semtomn Set of 2 Car Seat Covers Pink Animal of Schnauzer Dog Boy Portrait Cute Girl Universal Auto Front Seats Protector Fits for Car,SUV Sedan,Truck

Brand: Semtomn

Price: $37.90 & **FREE Shipping**

Get $60 off instantly: Pay $0.00 upon approval for the Amazon Rewards Visa Card. No annual fee.

Color: **Multi 1**

- Package: 2 Pcs front seat cover, armrest is NOT included.
- Easy to install, elastic strap with hooks on the bottom of the seat covers.
- Material: polyester. Soft and comfortable, easy to clean.

Figure 1: Examples of variations from Amazon catalog. Top-left: a normal variation listing, top-right: an inconsistent variation listing (incorrect item highlighted in the figure), bottom-row: an example of duplicate variations (two separate detail pages for the same product).

such as GTIN, UPC, EAN, etc., or similar attributes like model-number or model-name. However, the solution based on these attributes is not scalable, either due to sparsely available data or incorrect inputs. Information which are most reliably available in the catalog includes the product image, title and description. In this paper, we propose a scalable end-to-end multimodal approach to learning similarity between catalog items. We formulate it as a metric learning problem where the similarity between two items is reflected by the distance between their vector representations.

With the remarkable success from state of the art convolutional neural networks, many different applications have been developed leveraging image similarity such as recommendation, classification and search. However, the definition of similarity is very tightly coupled to their specific applications and may not be readily transferable to another application. Although the similar problem of fine-grained image recognition is quite extensively explored in academia, we extend the body of work to real world product catalog data for exact item matching without need for any explicit labels.

The main contributions of this paper are summarized as follows: 1) We present a scalable end-to-end multimodal framework for item similarity learning in a weakly supervised setting and discuss some of their important applications in the e-commerce domain, 2) The framework utilizes the product image and title to better generalize across multiple product categories, 3) We also propose a novel multimodal architecture for fine-grained image similarity task using transformer as an attention module, 4) We present several heuristics for training batch generation leveraging metadata from the item detail page.

The remaining of the paper is organized as follows: in the next section, we present some of the related works. Section 3 explains our approach in detail, followed by experimental results in section 4. We conclude with a summary and an outlook on future work.

2. Related Work

There is a lot of literature on product similarity matching with a majority of the approaches directed towards fashion images. These approaches have evolved from relying on traditional computer vision features [2, 3] to deep metric learning approaches with CNN-based features. Hadsell et al. [9] uses contrastive loss based on pair sampling in CNN to learn image similarity metric. Schroff et al. [16] successfully apply triplet loss to facial recognition tasks, which are highly similar to product similarity matching tasks. Studies have shown that visual attention is also adequate in many computer vision problems. Earlier work address this problem through learning discriminative features by localizing distinct regions of the image [26, 11]. However, these approaches need a labeled map for the localization of the distinctive regions and hence, is expensive and time-consuming. More recent contributions apply visual soft-attention techniques and achieved promising results. The attention is either applied spatially or channelwise. Wang et al. [23] propose to learn channel attention implemented by a fully convolutional network while Ak et al. [1] leverages activation maps to calculate attention map based on attribute classification results. These approaches primarily used only image based features for providing attention and solving the problem at hand. In e-commerce websites, there is a lot of textual information available apart from the image on the retail product page. These textual features like title, product category can be used to guide the attention in the images. In [6], Dong et al. use attribute aware spatial and channel attention to be able to locate relevant regions of interest for fine-grained fashion similarity. Li et al. [12] uses product image and metadata such as outfit popularity, title & category available on the fashion oriented websites to compose and score the fashion outfits automatically. Lin et al. [14] uses the product category information and present a scalable framework for guiding embeddings into different subspaces based on the category. [19], [5] and [13] takes multimodal approach for search, classification and retrieval.

3. Proposed Method

Formally, given an item a, let P_a denote its parent-item and U_a denote the base product identifier for the item. Ideally, P_a and U_a should have exact one-to-one mapping for all the items in the catalog. Our goal is to learn a representation, $f(a)$ such that: $d(f(a_1), f(a_2)) < d(f(a_1), f(a_3))$ where, $U_{a_1} = U_{a_2} \neq U_{a_3}$ for some distance measure, d and items a_1, a_2, and a_3. Now, optimizing the model with respect to U_a is difficult since it is generally unknown and figuring it out manually is costly. So instead, we take P_a as our ground truth labels and train the model in a weakly supervised fashion. Note that, using this notation, given two items a_1 and a_2, we can define, 1) DV as: $U_{a_1} = U_{a_2}$ & $P_{a_1} \neq P_{a_2}$ and 2) IV as: $U_{a_1} \neq U_{a_2}$ & $P_{a_1} = P_{a_2}$.

Next, we describe the various components of our framework for effectively learning the item representation using its image and title as the input. We first describe our model architecture which is inspired by [7]. In that paper, the author uses a fixed set of attributes to identify smaller regions-of-interest (RoI) in the image for learning attribute-specific embeddings. In this paper, we repurpose parts of the same architecture, while our main interest is to leverage the rich information which are present in the item title to mainly reduce the effect of background clutter in the image by focusing on the relevant regions. Further, we apply self-attention and introduce specific post-processing to detect larger RoIs from the input image and apply shallow layers to extract fine-grained features from the image [22]. Towards the end, we discuss some challenges and present few practical strategies associated with learning item embeddings on large-scaled unlabelled datasets and experiment with different loss functions and mining strategies.

3.1. Model Architecture

Consider an item a: (I, T) with image I and title T (*for simplicity, we will use a in place of (I, T) to denote the combined image and title input*). In our approach, we take the image as the primary source of product information, while the title helps the model focus on relevant regions in the product image. Our architecture, displayed in figure 2, extracts features from the image with both global view (global features $f_g(a)$) and local view (local features $f_l(a)$). The global features are generated from the whole image, whereas the local features are produced using the extracted region-of-interest from the original image. These features are then combined at the end to produce the final embedding of the item a.

The global branch f_g consists of an image feature extraction module which uses a convnet architecture along with self-attention[21] to convert the input image, I into a feature map, I_g. A spatial attention map is then computed by combining I_g and title features, F_t. Then a title attentive feature vector is computed by taking the weighted sum of I_g with the attention map. This feature vector is then passed through a channel attention module to produce the global feature $f_g(a)$. The spatial attention weights computed earlier are also used to crop a part of the input image after resizing it to the original input dimension, after applying specific post-processing operations. The cropped image is then passed through the local branch resulting in the local feature $f_l(a)$. Both of these branches are nearly identical and consists of an image feature extractor, spatial attention module and a channel attention module. The steps involved are visualized in Figure 4. Both global and local branches are trained using the procedure described in Sec 3.2.

A. Global Branch

The global branch serves two main purposes. First it is used to capture global features from the entire image. Second it is also used to crop out a relevant region or RoI from the input image which is then fed to the local branch. Here we describe in detail the components of the global branch. It consists of feature extraction modules for both image and title, a spatial attention module, and a channel attention module.

Feature Extraction Module Image: For image features extraction, I_g we use a modified version of the Deeprank architecture [22]. The original Deeprank architecture comprises of a VGG net along with two shallow convolution branches in parallel for fine-grained feature extraction. In our modified version, we replace the VGG with ResNet-50. We also add spatial self-attention on top of the Deeprank outputs. It enhances the features at each spatial location by adding the global context. Next we describe the self-attention module.

Self-Attention module: Let $R_g \in \mathbb{R}^{w \times h \times c}$ be the feature map from the last convolution layer of ResNet-50, where $w = 7$, $h = 7$, $c = 2048$ are the width, height and the number of channels in the feature map respectively. Similarly, let $S_1 \in \mathbb{R}^{w \times h \times 96}$ and $S_2 \in \mathbb{R}^{w \times h \times 192}$ be the outputs from the two shallow branches. These vertical slices at each spatial

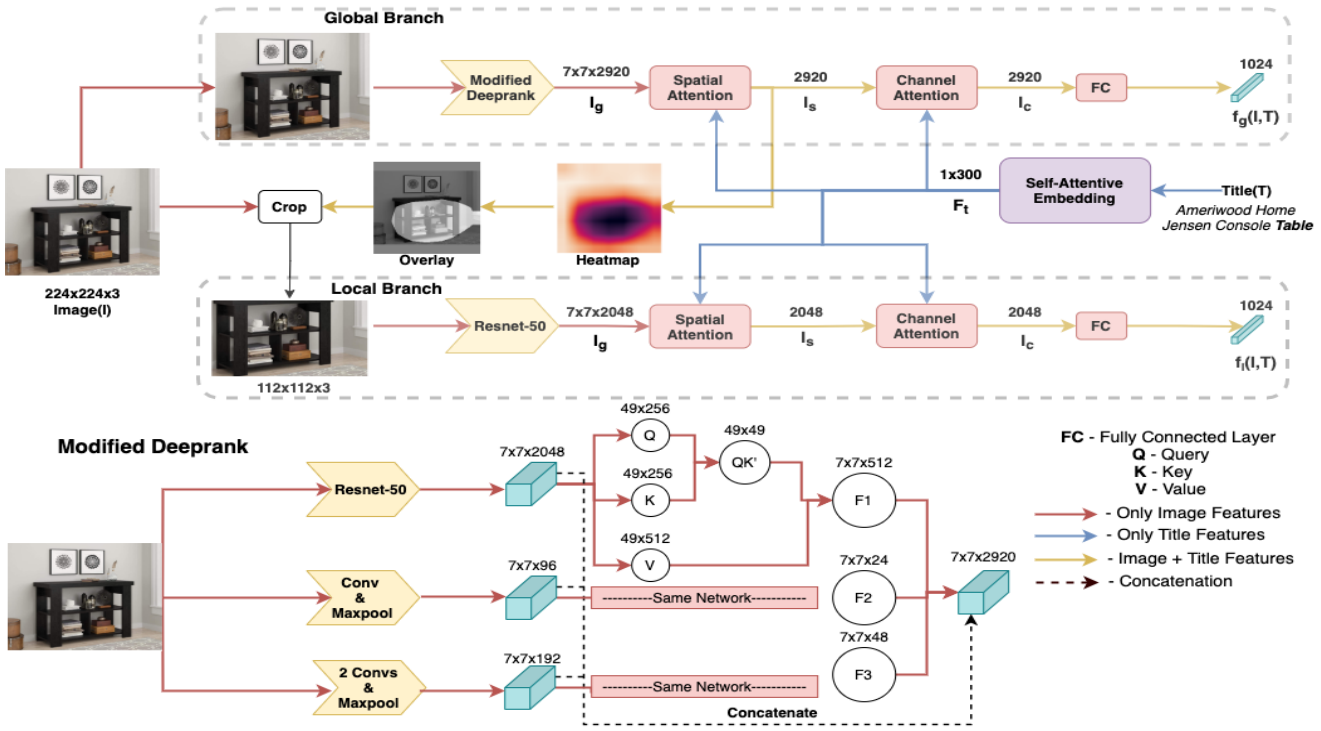


Figure 2: Proposed Network Architecture

location maps to different overlapping regions in the image and is limited by their respective receptive field. We apply self-attention on R_g, S_1 and S_2. It facilitates the spatial attention module to evaluate the relevance of each spatial regions in the context of the whole image. To apply self-attention, we flatten the output tensors along width and height axis resulting in feature matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\mathrm{m}]^T$, where $m = wh$. Each $\mathbf{x}_i$ is a feature vector of dimension c in case of R_g whereas it is 96 and 192 in case of S_1 and S_2 respectively. X is split into - query: $Q = XW_q$, key: $K = XW_k$ and value: $V = XW_v$, where $W_q, W_k, W_v \in \mathbb{R}^{c \times d_a}$ with d_a representing the depth of features in the attention mechanism. Final feature map after applying self attention on ResNet output R_g is calculated as [21]:

$$A_{\mathrm{R_g}} = softmax(\frac{QK^T}{\sqrt{d_a}})V \qquad (1)$$

where A_{Rg} denotes the final feature map from the ResNet50 output. Similarly, final feature maps from S_1 and S_2 is calculated using their respective query, key and values. All three features - A_{R_g}, A_{S_1} and A_{S_2} are then stacked, reshaped and concatenated with convolutional features to get I_g, which acts as the final image feature map of the global branch of dimension $7 \times 7 \times 2920$.

Feature Extraction Module Title: A typical title from an ecommerce detail page may look like this: *Emery Fabric Shower Curtain, 70"X70", Colorful Floral Geometric Printed Design*. It generally includes information such as the type of product and brand, and often also includes some additional information like size, color, style, model number, etc. In the above example, *Shower Curtain* is the type of product, *Emery* is the style, and *70"X70"* is the size. We hypothesize that the information from the title would help the model focus on relevant parts of the image. Each of these tokens does not contribute equally in identifying important regions in the image. In order to effectively make use of the information present in the title, we represent the title using self-attentive embedding [15]. To generate these embeddings, we first pre-process it by lowercasing all the words, removing non alphabetic characters, and then tokenize it using the FastText[8] dictionary. A 300 dimension embedding is then extracted for each token in the sentence from the dictionary. Next, we pass the sequence through a BiLSTM layer which produces a sequence of hidden states, $H = [H_1, H_2, \ldots H_n]$. The hidden states are then dotted with a weight vector, $\mathbf{w}_t$ to obtain a set of attention weights corresponding to each token. The LSTM hidden states are then averaged with these weights to get the final embedding of the title, F_t.

$$F_t = Hsoftmax(H^T \mathbf{w}_t) \qquad (2)$$

Spatial Attention Module: The spatial attention module combines the image and title features to identify the relevant regions in the image. As an example, suppose there is an image of a bedroom containing a bed, chair, table, etc. and the title contains the word chair in it. To extract useful feaures, we would want the model to emphasize more on the chair while ignoring the other objects in view. Given the image and title representations, I_g and F_t respectively, we obtain title attentive spatially weighted image features I_s. To compute I_s, we first project I_g to the same dimensionality as F_t using 1x1 convolutions. Then attention weights ($\alpha^s \rightarrow 7 \times 7$) are computed by applying Hadamard product between the transformed image and title features followed by softmax to normalize the weights. The final spatially attended image feature I_s is given by

$$I_{\text{s}} = \sum_{j}^{h \times w} \alpha^{\text{s}}{}_{\text{j}} (I_{\text{g}})_{\text{j}} \tag{3}$$

Channel Attention Module: The channel attention module helps filter out dimensions in the spatially attended image features, I_s which are not relevant for item comparison. For e.g., the color token, *Red* in the title - '*Torque - Dalton 6 Seater L Shape Corner Sofa for Living Room (Right Side, Red)*', can help locate a red coloured sofa in the image. Once located, we don't want to retain color specific features in the final representation. For this, the title feature, F_t is first projected to a new dimension using a fully connected layer. This transformed feature vector and the spatially attended image features, I_s are then concatenated and fed into two fully connected layers to produce the channel attention weights α^c, having the same dimensions as that of I_s. To get the final output (I_c), Hadamard product is computed between I_s and α^c.

After getting the output (I_c) from the channel attention module, we employ a fully connected layer on I_c to generate the final title (T) attentive global feature of the image $I \rightarrow f_g(I, T)$:

$$f_g(I, T) = f_g(a) = WI_c + b \tag{4}$$

where $W^{c_o \times c}$ is the transformation matrix, c is the size of I_c, c_o is the output dimensionality and b is the bias term and a is the item.

B. Local Branch

The local branch uses the cropped RoI, from the original image, to look at the item more closely to extract more relevant features. In order to identify the RoI, the attention map, α^s computed earlier is first resized to the original input size (224x224). It is then binarized using a threshold to obtain a binary map. Connected component analysis is applied on the binarized map to filter out the spurious regions from the map. A bounding box is then extracted from this binary map to produce the cropped region (RoI) from the input image. This cropped region is padded with black pixels to produce a square image of size 112x112, which acts as input image to the local branch. This input image is passed through the ResNet-50 block instead of the Deeprank based architecture used in the global branch. Rest of the details (feature extraction for title, spatial attention and channel attention) are similar to that of the global branch giving us the final title (T) attentive local feature of the image $I \rightarrow f_l(I, T)$ or $f_l(a)$.

3.2. Training and Batch Sampling

Given an item $a : (I, T)$, with image I and the title T, let $f_g(a) \in R^d$ and $f_l(a) \in R^d$ be the d dimensional feature embeddings returned by the global and local branch respectively. Let us consider two items, a_1 and a_2. Now, since we are considering parent-item as the ground truth label, if $P_{a_1} = P_{a_2}$, we want to have global features, ($f_g(a_1)$ and $f_g(a_2)$) as well as local features, ($f_l(a_1)$ and $f_l(a_2)$) closer together. However, if $P_{a_1} \neq P_{a_2}$, we would want to have these embeddings further apart from each other.

Loss Function: Minimizing Triplet Loss([10, 17]) and NTXent Loss([20, 4]) are effective techniques for learning these representations. Triplet loss works by minimizing the distance between the anchor (p_a) and positive (p_p), and maximizing the distance between anchor and negative (p_n) such that it satisfies the margin constraint α.

$$Loss_{triplet} = [d(p_a, p_p)^2 - d(p_a, p_n)^2 + \alpha] \tag{5}$$

Triplet loss optimizes a 3-tuple (p_a, p_p, p_n), where it pushes a single negative sample away from the positives one at a time. Whereas the NTXent loss, represented by the equation below optimizes an n-tuple $(p_a, p_p, p_{n_1}, \ldots, p_{n_{(N-2)}})$, where it pushes away multiple negative samples simultaneously.

$$Loss_{ntx}(p_a, p_p) = -\log \frac{\exp(-d(p_a, p_p)/\tau))}{\sum_{k=1}^{N} 1_{k \neq a} \exp(-d(p_a, p_k)/\tau)} \tag{6}$$

Here p_a and p_p are the positives and share the same label. All the other k's in the denominator are negative samples w.r.t. to a.

Batch Sampling: In this work, anchor and positive are sampled from the same parent ($P_{p_a} = P_{p_p}$), and negatives are sampled from separate parents ($P_{p_a} \neq P_{p_n}$). Randomly created tuples easily satisfy the loss constraints, and thus are not very useful for learning. Hence several hard mining strategies have been proposed in the literature [24]. The idea of hard mining is to sample only those tuples for training which are difficult to optimize. As such, these mostly includes cases where anchor and negatives are sampled from similar products. Searching hard triplets (or any n-tuple) in large datasets is computationally expensive, hence in practice, these are sampled from within a batch [18]. In practise, this requires a sufficiently large batch size so as to have enough contrastive samples for effective learning. This is especially true in cases where the dataset consists of a large number of product categories. To ensure faster convergence with smaller batch, we leverage product details such as item-brand and item-subcategories to group together items of similar types to create a batch. We find that this simple approach results in significant improvement in the performance. We compare the results of using different types of keywords for batch creation in Table 5 (Sec. 4.4).

Training: We apply one of these losses (triplet or NTXent loss) on both the global(L_g) and the local branch(L_l). We also include an alignment loss(L_a) between global and local features, which forces the model to embed the original image and its localized version consistently. This loss is given by the distance between the global feature $f_g(a)$ and the local feature $f_l(a)$. Following [7], we adopt a two-staged procedure for training our model. The first phase optimizes only the global branch using the L_g loss. Once the global branch stabilizes, both the branches are jointly trained in the second stage:

$$L = w_g L_g + w_l L_l + w_a L_a \tag{7}$$

where w_g, w_l, and w_a are the hyperparameters which control the importance of the different losses.

Inference: During inference, the distance between a pair of items, a_1 and a_2, is computed as the weighted sum of the L2 distances between their global and local features:

$$d(a_1, a_2) = \lambda d_{l_2}(f_g(a_1), f_g(a_2)) + (1-\lambda) d_{l_2}(f_l(a_1), f_l(a_2)) \tag{8}$$

Here λ is the hyperparameter to balance the importance of 2 features(global and local), and $d_{l_2}(p, q)$ is the L2 distance between the two feature vectors p and q.

4. Experiments

In this section, we report experimental results testing the effectiveness of the proposed framework, starting with the dataset description.

4.1. Datasets

The dataset used in this work is obtained from the Amazon catalog and consists of three product categories. For training data creation, we fetched random parent-items (families) along with all the child-items from the product detail page. For each child-item, we extracted the following details - image, brand, title and, subcategory (subcat). Additionally, we filter out items with duplicate images using the image-id so that all the images in the dataset are unique. During testing, we want to measure how well the model is able to distinguish between similar and dissimilar items. Therefore, each test data sample consists of item pairs that can either be from the same (S) or different (D) product (ref. figure 3). To create the test data, we randomly paired up items from across two different families and labelled them manually by comparing their images. Details of the datasets used are provided in Table 1.

Category	**Train Size**	**Test Size**	
		Same Product	**Different Product**
A	43447	5508	10352
B	57346	7758	13831
C	114295	9645	11407

Table 1: Details of train and test dataset

4.2. Evaluation Metrics

Since every pair in our test set belongs to either the S or D class, it is a binary classification task. In order to make the prediction, we compute the distance between the items in a pair and compare it with a pre-defined threshold. We use three different metrics for evaluation, PR-AUC, Precision@Recall=50, Precision@Recall=75.

4.3. Implementation Details

We ran the experiments in PyTorch. The training time is approximately 20 hours on a single Nvidia V100 GPU with a batch size of 48. The dimensions of the input image were kept as 224x224x3 for the global branch while it was fixed to 112x112x3 in the local branch. The title was preprocessed by lowercasing all the words and removing numbers from the title. It was then tokenized using the fasttext dictionary. The modified deeprank backbone network in the global branch returns a 2336-D feature vector for the image while the title is represented using 300-D glove embeddings. ResNet-50 was used as a backbone network in the local branch, and it returns a 2048-D feature map. Our final embedding dimension for both global and local feature networks is 1024. The global branch was first trained for 2000 steps, followed by the whole network training for 30000 steps. The loss functions weights w_g, w_l and w_a were kept as 1. λ was empirically determined and was found to be 0.55.

4.4. Results

First, we test the model's scalability by comparing the performance of a single model trained across all three product categories vs individual models trained separately on each category. We hypothesize that since the model uses title for attention, it will be able to utilize this context by focussing on the relevant region or features from the image. The results are shown in Table 2. The first row displays the results corresponding to separate models trained on individual

Category	A			B			C		
Model type	**P@50R**	**P@75R**	**AUC**	**P@50R**	**P@75R**	**AUC**	**P@50R**	**P@75R**	**AUC**
Individual	91.86	87.48	89.03	92.12	77.93	85.92	91.18	85.31	88.40
Combined	90.47	86.35	87.54	93.76	79.23	85.84	91.01	84.24	87.62

Table 2: Comparison of models trained on individual datasets with the model trained on combined dataset.

datasets (eg. the first 3 columns of the first row (*Individual*) show the results corresponding to the model trained only on category A). The 2nd row (*Combined*) reports the model's performance for the combined training.

We observe a slight degradation ($\approx 1\%$ drop in Precision@75R) on both category A and C as a result of combined training, while performance on the category B remains the same. This results in improved scalability and will help reducing model proliferation.

Baseline Comparison: Next, we evaluate our approach on the combined dataset against ResNet-50, Deeprank and a recent multimodal approach, [7] in table 3. The first two baselines take only image as the input, whereas the multimodal baseline (row 3) uses both image and product type keyword as input. For ResNet-50, we took the output from the final layer as the image feature (2048-D). Compared to the two image based baselines (rows 1 & 2), we observe significant improvement with our multimodal approach (row 4). The proposed approach also performs better than the multimodal baseline on all three product categories.

Ablation Experiments: We perform ablation studies to show the importance of different components in our proposed framework. In the first experiment, we provided the attention using subcat instead of using the product title. As subcat is generally composed of short phrases (such as *running shoes*, *t-shirts*), we averaged fasttext embedding of those tokens and used it in place of the title embeddings. The results are reported in the row *Proposed (Attn. subcat)* in Table 3. 1-2% drop can be observed in almost all the cases compared to *Proposed (Attn. title)*. We also conducted ex-

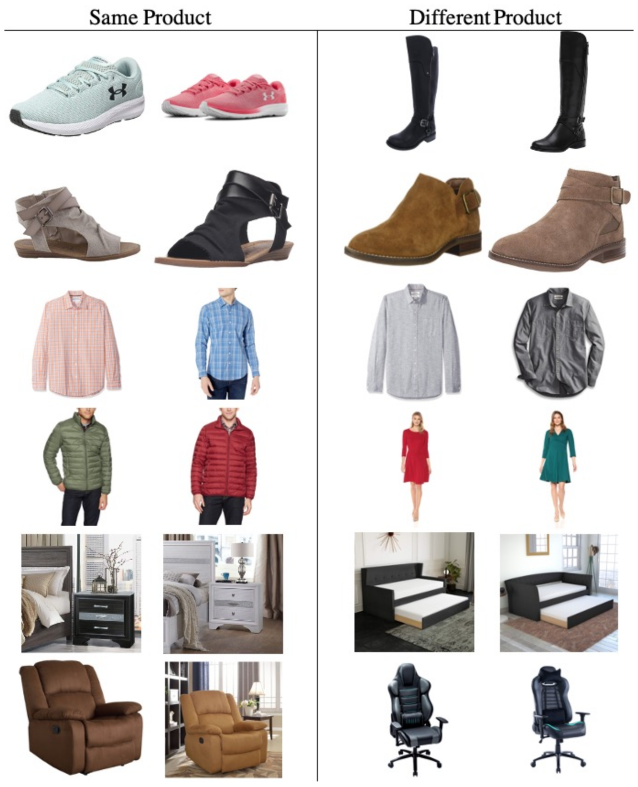


Figure 3: Examples of successful predictions by the model. Left column: Same products predicted as same. Right column: Different products predicted as different.

Category	A			B			C		
Model type	**P@50R**	**P@75R**	**AUC**	**P@50R**	**P@75R**	**AUC**	**P@50R**	**P@75R**	**AUC**
Baseline ResNet	72.97	61.01	72.63	85.49	66.26	79.63	81.86	69.81	80.43
Baseline Deeprank	83.15	74.55	80.73	88.04	69.16	81.15	83.40	79.47	83.74
Dong et al. [7]	85.07	75.83	82.33	90.13	75.69	83.78	87.32	81.66	85.87
Proposed(Attn title)	**90.47**	**86.35**	**87.54**	**93.76**	**79.23**	**85.84**	**91.01**	**84.24**	**87.62**
Proposed (Attn subcat)	87.93	80.04	84.92	92.85	76.49	84.25	87.16	80.44	85.91
Proposed w/o SA	88.27	83.53	86.25	93.48	77.31	84.37	87.87	82.01	86.28
Proposed w/o CA	89.85	85.28	86.89	93.53	78.48	84.80	88.55	82.34	87.03

Table 3: Comparison with baselines (1st part) and ablation experiments (2nd part)
Abbr. - SA: Self Attention, CA: Channel Attention, Attn: Attention, w/o: without

Category	A			B			C		
Loss/Mining	**P@50R**	**P@75R**	**AUC**	**P@50R**	**P@75R**	**AUC**	**P@50R**	**P@75R**	**AUC**
NTXent All	82.25	76.54	81.94	86.73	75.33	83.08	85.77	78.00	85.28
NTXent EPSHN	**90.47**	**86.35**	**87.54**	**93.76**	**79.23**	**85.84**	**91.01**	**84.24**	**87.62**
Triplet All	86.22	77.44	83.53	89.64	73.68	83.70	85.58	80.09	85.37
Triplet EPSHN	84.76	76.31	82.69	90.48	78.15	84.58	84.81	79.53	84.64

Table 4: Comparison of loss functions and mining strategies

periments to understand the importance of the self-attention and the channel-attention module described in Sec 3.1 by dropping the respective modules from the architecture. The results are reported in the last two rows in Table 3.

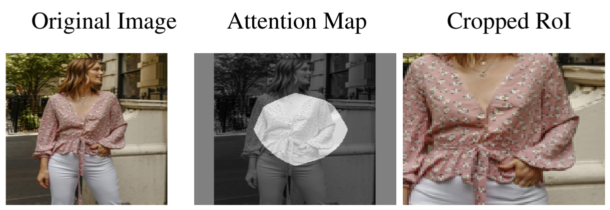
We also experimented with different loss functions and mining strategies. More specifically, we trained our architecture using two different mining techniques - i) All mining and ii) Easy Positive Semihard Negative mining (EPSHN [25]) and two different loss functions - i) Triplet Loss and ii) NTXent loss [20]. The results are reported in the Table 4. We observe that combination of NTXent loss and EPSHN mining gives overall the best results. Surprisingly, triplet loss performs slightly better with All mining strategy compared to EPSHN mining.

As discussed in section 3.2, we also compare brand, subcat and brand+subcat as grouping keywords for batch creation. If we do not apply any grouping, then very few batches have informative tuples, resulting in a longer training duration. We find that the combination of brand+subcat results in the batch which is most homogeneous and thus gives the best results. The averaged results across all three product categories are reported in the table 5 below.

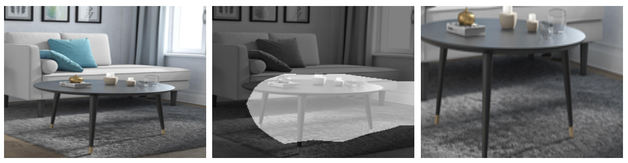
Keyword Used	P@50R	P@75R	AUC
subcat	66.71	54.10	66.29
brand	80.43	73.89	82.18
brand+subcat	**91.75**	**83.27**	**87.00**

Table 5: Comparison of few keywords for batch creation

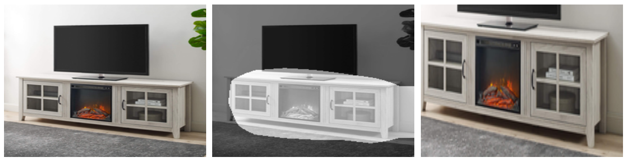
Visual Analysis: Visualization of the process of getting the RoI, starting from the original input image to getting a cropped RoI from it, are shown in Fig 4. The first image in each row is the input image to the global branch of the network. After processing by the global branch, as described in Sec 3.1, we get an attention map highlighting the important regions in the image. This attention map is then thresholded, binarized, and resized (2nd image is overlay of resized attention map on the original image) to get a cropped RoI (3rd image) from the original image. This cropped RoI is then passed as an input to the local branch of the network. The product title is displayed below the images. Also, in figure 3, we visualize few of the predictions by our model.



Title: Drop Women Ecru Draped Pleated Cross Front Volume Sleeve Button Shirt Lisadnyc



Title: Kennington Coffee Table, Black



Title: Walker Edison Bern Classic 2 Glass Door Fireplace TV Stand for TVs up to 80 Inches, 70 Inch, Birch

Figure 4: Getting cropped RoI using product title

5. Conclusion

In this work, we presented an end-to-end multimodal framework for learning product similarity, leveraging both the product image and title. We also talked about some important applications of learning product similarity in ecommerce domain, and highlighted the challenges involved in training such models on real world unlabelled datasets. We demonstrated the benefit of incorporating multimodal attention in terms of improved scalability and performance. We also observe good improvement (up to 5.2% in AUC) in the performance compared to multimodal (image + text) state of the art. Possible future directions of the current work include incorporating additional details such as product description, bullet-points and other additional attributes to allow for more finegrained comparison, and leveraging self-supervised pretraining to better capture the context between the image and textual domains.

References

[1] Kenan E. Ak, Ashraf A. Kassim, Joo Hwee Lim, and Jo Yew Tham. Learning attribute representations with localization for flexible fashion search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, 2010.

[3] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *J. Mach. Learn. Res.*, 11:1109–1135, Mar. 2010.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[5] N. Das, D. Mandal, and S. Biswas. Simultaneous semi-coupled dictionary learning for matching in canonical space. *IEEE Transactions on Image Processing*, 26(8):3995–4004, 2017.

[6] Jianfeng Dong, Zhe Ma, Xiaofeng Mao, Xun Yang, Yuan He, Richang Hong, and Shouling Ji. Fine-grained fashion similarity prediction by attribute-specific embedding learning, 2021.

[7] Jianfeng Dong, Zhe Ma, Xiaofeng Mao, Xun Yang, Yuan He, Richang Hong, and Shouling Ji. Fine-grained fashion similarity prediction by attribute-specific embedding learning, 2021.

[8] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[9] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.

[10] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification, 2017.

[11] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks, 2016.

[12] Yuncheng Li, Liangliang Cao, Jiang Zhu, and Jiebo Luo. Mining fashion outfit composition using an end-to-end deep learning approach on set data. *IEEE Transactions on Multimedia*, 19(8):1946–1955, Aug 2017.

[13] Lizi Liao, Xiangnan He, Bo Zhao, Chong-Wah Ngo, and Tat-Seng Chua. Interpretable multimodal retrieval for fashion products. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 1571–1579. Association for Computing Machinery, 2018.

[14] Yen-Liang Lin, Son Tran, and Larry S. Davis. Fashion outfit complementary item retrieval, 2020.

[15] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[16] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

[17] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.

[18] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

[19] Ivona Tautkute, Tomasz Trzciński, Aleksander P. Skorupa, Łukasz Brocki, and Krzysztof Marasek. Deepstyle: Multimodal search engine for fashion and interior design. *IEEE Access*, 7:84613–84628, 2019.

[20] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[22] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[23] Zhonghao Wang, Yujun Gu, Ya Zhang, Jun Zhou, and Xiao Gu. Clothing retrieval with visual attention model, 2017.

[24] Hong Xuan, Abby Stylianou, and Robert Pless. Improved embeddings with easy positive triplet mining, 2020.

[25] Hong Xuan, Abby Stylianou, and Robert Pless. Improved embeddings with easy positive triplet mining. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2463–2471, 2020.

[26] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification, 2018.