Learning Disentangled Multimodal Representations for the Fashion Domain

Amrita Saha*, Megha Nawhal+, Mitesh M. Khapra†, Vikas C. Raykar*
*IBM Research, India, {amrsaha4, viraykar}@in.ibm.com
+Simon Fraser University, Canada, mnawhal@sfu.ca
† IIT Madras, India, miteshk@cse.iitm.ac.in

Abstract

In many visual domains (like fashion, furniture, etc.) the search for products on online platforms requires matching textual queries to image content. For example, the user provides a search query in natural language (e.g.,pink floral top) and the results obtained are of a different modality (e.g., the set of images of pink floral tops). Recent work on multimodal representation learning enables such cross-modal matching by learning a common representation space for text and image. While such representations ensure that the n-dimensional representation of pink floral top is very close to representation of corresponding images, they do not ensure that the first k_1 ($< n$) dimensions correspond to color, the next k_2 ($< n$) correspond to style and so on. In other words, they learn entangled representations where each dimension does not correspond to a specific attribute. We propose two simple variants which can learn disentangled common representations for the fashion domain wherein each dimension would correspond to a specific attribute (color, style, silhoutte, etc.). Our proposed variants can be integrated with any existing multimodal representation learning method. We use a large fashion dataset of over 700K fashion items crawled from multiple fashion e-commerce portals to evaluate the learned representations on four different applications from the fashion domain, namely, cross-modal image retrieval, visual search, image tagging, and query expansion. Our experimental results show that the proposed variants lead to better performance for each of these applications while learning disentangled representations.

1. Introduction

In many visual domains (like fashion, furniture, jewellery, etc.), the search for products on online platforms is highly driven by *visual attributes*. By visual attributes we mean the properties of the product (apparel in the fashion domain) which eventually decide how it looks. For example, a typical query on a fashion e-commerce site can be *'pink floral print top'*. The query generally consists of the apparel or accessory the user is looking for (*top*) along with various visual attributes like color (*pink*), pattern (*floral print*), fabric, silhouette, etc.

To enable such search, one approach is to manually curate and tag all these visual attributes for every product in the catalog. Once we have these attributes then the search process is driven by a suitable indexing mechanism to retrieve the relevant products either via the search query or the various attribute filters. In other words, the search problem reduces to matching the text in the query to the text in the tags. However, all visual attributes may not be available for all the products in the catalog. For instance, a dress may have the color attributes but not have the pattern attributes. Moreover, manually annotating all the visual attributes for every item in the catalog is not be scalable. While this is possible for catalogs with limited items, modern fashion portals have huge catalogs. Also, for aggregators, these attributes are not standardized across different vendors and all attributes may not be annotated for all items.

Given the scalability issues with tagging and searching, we propose to drive the catalog search by learning a common representation for the image and its corresponding text description (see Figure 1). A representation for an image or a query text is a fixed dimensional vector that encodes as much semantic information as possible. For example, the query phrase *'pink floral print top'* can be encoded into a fixed dimensional vector. Similarly, the corresponding image of the *top* would be encoded into another fixed dimensional vector. Further, a joint representation learning algorithm [24, 34, 31, 26, 9, 8, 23] would ensure that these two representations are in the same space and as close as possible to each other. If all the images in the catalog are represented in this common space and the text query is also represented in this space then the search operation reduces to finding the images which are closest to the query in this common space.

While the above method works well in practice, we observe that existing algorithms for multimodal representation learning learn entangled representations. In other words, if the image/text is represented by a n-dimensional representation, it is not clear what each dimension encodes. In specific

domains, such as fashion, where we know that both the image and query are composed of specific attributes, *viz.*, color, style, pattern, apparel, fabric, etc., it is desirable to learn disentangled representations such that certain dimensions in the representation correspond to certain attributes. For example, the first k_1 dimensions correspond to color, next k_2 to style and so on. This not only makes the representations more interpretable but also leads to more robust learning. For example, this would ensure that the first k_1 dimensions in the joint representation of the text *'pink floral print top'* and its image correspond to *'pink'*, the next k_2 dimensions to *'floral'* and so on. We pick one such joint representation learning method, *viz.* CorrNet [4] (see § 3), and propose two simple variants which allow us to learn such disentangled representations. The proposed variants can be integrated with any such joint representation learning method but in this paper we only focus on CorrNets. To learn these representations, we have curated a large fashion dataset of around 700K images along with description crawled from multiple fashion e-commerce portals (see § 5).

We evaluate the learned representations on four different applications in the fashion domain, *viz.*, text based image retrieval, visual query based image retrieval, image tagging, and query expansion. The retrieval mechanism works as illustrated in Figure 1. Every image in the catalog is first passed through our model to generate a fixed dimensional representation. We then create a index for all images in the catalog for fast nearest neighbor retrieval. For any query phrase, we extract the corresponding representation (which is in the same space as the image representations) and find the nearest neighbor images as the relevant results. Since the images and the description lie in the same space, the same index can also be used for visual search where the query is an image and the output is a set of images relevant to the image query. Essentially, the same model and the index can be used both for text and visual search as illustrated in Figure 1. An added advantage of this approach is that instead of images one could also build the index using the representation for all the tags in the product catalog. In this mode if the input is an image then the output would be a ranked list of tags or descriptions relevant to the image as illustrated in Figure 1. Similarly, if the input is a text query then the output is a list of relevant descriptions similar to the user query, which could be useful for query expansion. Through our experiments, we demonstrate that for all these applications the proposed variants clearly outperform vanilla CorrNet and other strong baselines.

2. Related work

Learning attributes There has been some work in the computer vision community in the area of learning visual attributes. These ideas have been adapted to the fashion domain to automatically recognize and classify people's clothing in natural scenes and explicitly predict the various visual attributes [2, 6, 5, 32]. These approaches typically learn a separate classifier for different apparel types and for each possible attribute. For example, [2] based the system on 15 clothing categories and 8 attribute categories with a total of 78 attributes. The main drawback of these methods is that one has to carefully curate a large labeled set of images for different clothing and attribute types which may not be scalable for large number of attributes and will be harder to maintain when new attributes get added. Each of these classifiers has to be applied to the image to predict the attribute values. We take an alternate approach to the problem based only on representations; while we do not explicitly predict the apparel category and the visual attributes for the image, the representations learned using our approach encode all this information. As illustrated in Figure 1 our proposed model can also be used to predict the attributes.

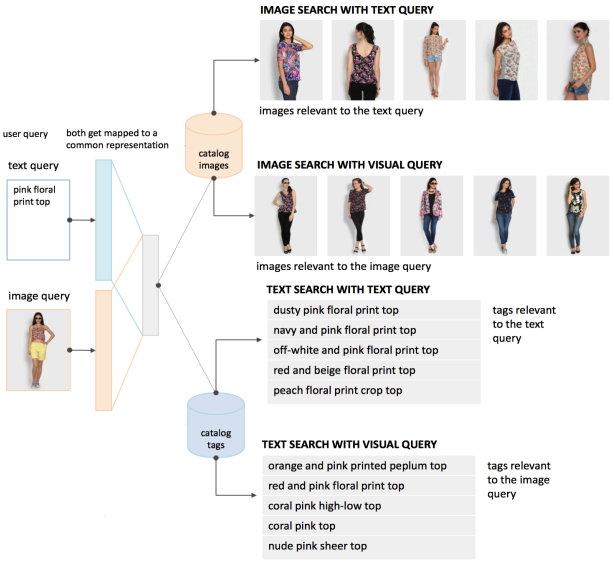

Figure 1. Multimodal representations for catalog image/tag search

Visual Search There is another line of work where the search is driven by images [13, 14]. Current state-of-the-art systems typically use an intermediate representation(usually the *fc6* layer) from a deep Convolutional Neural Network (CNN) trained for that domain. While our focus in this work is to learn representations for search phrases, since we are learning joint representations we can also do visual search using the representations for the image. We also start with the representation from the CNN and further encode it to lie in the same space as the corresponding description of the image. In the experimental section, we demonstrate how we can do a better visual search using our representations since the textual descriptions help us learn a better representation for visual search.

Query representations While distributed representations for word and phrases (*e.g.*, word2vec [16] or GloVe [22]) exist, they will not be relevant for our approach unless they

are grounded in the same space as the representations for images. These representations are generally trained to predict the other words in the context however in this case we train these representations using the visual input from the images. In the experimental section, we show some examples comparing our representations to those from word2vec.

Multimodal representations Canonical Correlation Analysis (CCA) and its variants [10, 29, 19, 7, 1] are the most commonly used methods for learning a common representation for two views. In the specific case of Multimodal Representation Learning each view belongs to a different modality (audio, video, image, *etc*). Recently Neural Network based approaches have become popular for learning common representations for multiple modalities [11, 17, 30]. For example, [17] proposed an autoencoder based solution to learning common representation for audio and video. [27] extended this idea to RBMs and learned common representations for image and text. Other solutions for image/text representation learning include [35, 36, 25]. In this work, we use Correlational Neural Networks (CorrNet) [3] which can learn common representations for multiple views using an encoder/decoder approach with a correlation based regularizer. We found this to be easily adaptable to our application as discussed in the next section.

[24, 34, 31, 26, 9, 8] are various other contemporary deep learning architectures for joint representation learning, which are comparable in performance to CorrNet. [24, 34] are based on enhancements over the Canonical Correlation Analysis (CCA) technique, which has been shown to be beaten by CorrNet by a much larger margin for an extensive set of tasks in [3]. The other contemporary work using stacked auto-encoders [31] is very similar both in terms of architecture and performance to the CorrNet and its deeper variant as studied in [3]. Our goal in this paper is not to evaluate various multimodal embedding techniques, but rather choose a simple one that performs well and apply it to the domain of fashion e-commerce. We used CorrNet because it is one of the state-of-the-art and both easy and fast to train. [9] is another work which deals with modalities other than the ones of interest for fashion or e-commerce search. Both video and auditory representation learning is very different from learning common representations between images and text hence these works are not comparable or applicable to our setting. [8] is another recent work in the related area, but with a very different focus; that of learning multilingual representations by leveraging an anchor image connected to captions of different languages.

Another competing work is [26] whose performance is very close to the deep autoencoder model in [18]. While these models focus on leveraging multimodal features for end applications, CorrNet learn a common representation that captures the low-level features to enable effective cross-modal applications, even when one of the modalities is missing. The performance of both these methods are very similar to that of CorrNet across multiple tasks, as has been shown in experimental results of [18] and there is no strong empirical evidence of the superiority of either of these techniques. Even though their architecture is very similar to CorrNet, there are clear differences in the training procedure. Unlike CorrNet model, the objective function used in the other methods do not enforce the network to learn the correlation while learning the common representations.

3. Background: Correlational Neural Network

In this section we describe the Correlational Neural Network (CorrNet) [3] used for learning common representations for two views, *viz.*, catalog descriptions of fashion items and their images. Each image (denoted by X) is represented using the 4096 dimensional (denoted by d_X) representation obtained from the *fc6* (fully connected layer 6) of a pretrained ConvNet (BVLC CaffeNet [12]). The catalog descriptions (Y) are represented using a bag-of-words feature vector of dimension d_Y.

Given such two-view data $x_i, y_i{}_{i=1}^{M}$ consisting of M $\{text(x), image(y)\}$ pairs, CorrNet learns encoder functions to compute a hidden representation of the two views. Let $h_X(x_i) \in \mathbb{R}^k$ and $h_Y(y_i) \in \mathbb{R}^k$ be the respective encoded projections of x_i and y_i in a common subspace $\mathbb{R}^k$.

$$h_X(x) = f(Wx + b), \quad h_Y(y) = f(Vy + b) \quad (1)$$

where f can be any non-linear activation function like sigmoid or tanh, $W \in \mathbb{R}^{k \times d_X}$ and $V \in \mathbb{R}^{k \times d_Y}$ are the encoder matrices for views X and Y respectively and $b \in \mathbb{R}^k$ is the bias vector. For simplicity, hereafter $h(x)$ also denotes $h_X(x)$. Similarly, CorrNet also has a decoder for each view as follows which tries to reconstruct the original input from the hidden representations:

$$z_X(h) = g(W'h + c_X), \quad z_Y(h) = g(V'h + c_Y) \quad (2)$$

where g can be any activation function, $W' \in \mathbb{R}^{d_X \times k}$ and $V' \in \mathbb{R}^{d_Y \times k}$ are the decoder matrices for views X and Y respectively and $c_X \in \mathbb{R}^{d_X}$ and $c_Y \in \mathbb{R}^{d_Y}$ are the bias vectors for both views respectively. In addition, we also define an encoder which works on the concatenated representation $o = (x, y)$ and encodes it as $h_O(o) = f(Wx + Vy + b)$. Given the above setup, CorrNet then tries to minimize the error in reconstructing the original input (x, y or the concatenated representation o) from either of the following hidden representations, h_X, h_Y of h_O while maximizing the similarity between $h_X(x_i)$ and $h_Y(y_i)$ (i.e., ensuring that corresponding x_i and y_i have very similar representations).

3.1. Limitations of the CorrNet architecture

In fashion domain, there are certain attributes which are more important than others. In particular, the color, texture and apparel in the image and the corresponding words

describing color, pattern and apparel in the text are most important for characterizing a product. Ideally, the joint representation learning algorithm should be aware of these and try to ensure that these attributes from the image and text are aligned in the common representation. For example, we would want that certain dimensions in the compact representation of the image capture the color information, certain dimensions capture the texture information and so on. Similar argument can be made with respect to the hidden representations of the textual description. Further, the joint representation learning algorithm should ensure that corresponding dimensions in the hidden representations of the two views are correlated. In particular, the dimensions capturing the color information in the image representation should be correlated with the dimensions capturing the color information in the text. In other words, we want to learn representations which are disentangled for different attributes within a view and correlated for corresponding attributes across views.

The above objectives are indeed hard for CorrNet to achieve because it does not get any explicit supervision to decompose the task into *'align-and-correlate'* steps. We propose a two step approach instead, wherein in the first step we learn to align the representations of each attribute in the text independently with the image. In the second step, we learn to correlate the representations of the entire text with the image while respecting the alignments learned earlier.

4. Proposed Models

In this work, we consider three important attributes from the fashion domain, *viz.*, color, pattern and apparel. We propose two variants of CorrNet to learn disentangled representations with respect to these attributes. The first variant is just a trivial extension of CorrNet which works very well in practice whereas the second variant is a more principled approach which builds on the first one. We describe these two variants in the following sub-sections.

4.1. Attribute Based CorrNet

As discussed above, we want that the dimensions encoding information about the *color* mentioned in the text should align with the dimensions encoding the information about *'color'* in the image. One simple way of doing this is to prune all the words in the textual description except the *'color'* words and then learn a joint representation for the text and the image. For this, we collect a vocabulary of all the colors in the dataset. We then create {image, text} pairs where the text contains color words only and subsequently train a CorrNet which will try to learn a joint representation for the text (color) and image as described earlier. However, now the textual representation is clearly disentangled from the other attributes as it only contains information about one attribute (color, in this case). But how do we ensure that the image also contains only color information? In other words, how do we prune out everything except the color of the apparel in the image? This is clearly hard to do in the absence of explicit annotations on the image. Instead, we use the fc6 representation of the full image as before. However, now since there is a constraint on the learned hidden representation to correlate with the color words, it is conceivable that the hidden representation will sufficiently encode information about the color. Indeed, we observe that at the end of training, if we compute the representation of the word *'red'* and then fetch all images whose representation are closest to this representation then we get all red colored apparels. Similarly, we train two more attribute based CorrNets, one for the *'apparel'* and one for the *'pattern'*. Once again, we observe that at the end of training the model ensures that the representation of the word *skirt* is closest to all images containing skirts.

4.2. Align-and-Correlate CorrNet

Though the attribute based CorrNet model does well in various end applications, it requires a post-processing step to merge results from the individual CorrNets. This step often needs external domain knowledge. For example, it needs to know that it is more important to fetch images which match the *'apparel'* in the query than fetching images which match the *'color'* or the *'pattern'*. In many cases, such domain knowledge may not be known ahead of time, so it is desirable that the model learns the importance of different attributes in the domain and ranks the final results accordingly. Hence, a more principled approach would be to have a single model which handles all the three attributes together. Such a model would be trained using parallel text-image data, where the text part is a concatenation of the bag-of-apparel-words, bag-of-color-words and bag-of-pattern-words whereas the image part is the same ConvNet *fc6*-layer representations as used earlier. Intuitively, it would help if this model exploits the attribute specific joint representations which were already learned by the three attribute based CorrNets. To do so, we propose to train a CorrNet model whose encoder and decoder parameters are initialized in the following way:

$$W_0 = \begin{pmatrix} W_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & W_c & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & W_p \end{pmatrix}, \quad V_0 = \begin{pmatrix} V_a \\ V_c \\ V_p \end{pmatrix} \tag{3}$$

$$b_0 = \begin{pmatrix} b_a \\ b_c \\ b_p \end{pmatrix}, \quad c_{0_X} = \begin{pmatrix} c_{1_a} \\ c_{1_c} \\ c_{1_p} \end{pmatrix}, c_{0_Y} = c_{Y_a} + c_{Y_c} + c_{Y_p} \tag{4}$$

where the parameters corresponding to the three individual attribute based CorrNets are indicated by the appropriate subscripts, i.e., subscript *'a'*, *'c'* and *'p'* for the *'apparel'*, *'color'* and *'pattern'* CorrNet respectively. **0** denotes a matrix

of 0s of a compatible shape. Subscript 0 in W_0, b_0 etc. denotes the initial state of the corresponding tensors.

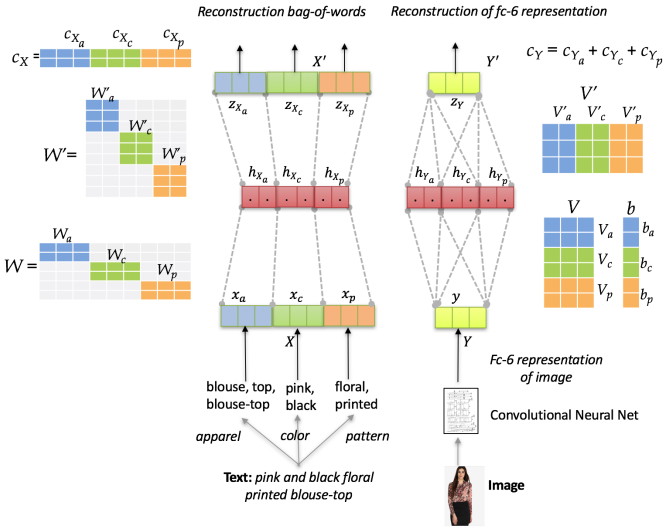

Figure 2. Pretraining of the Correlational Neural Network with the alignment over the different salient attributes. The *'blue'*, *'green'* and *'orange'* represent the *'apparel'*, *'color'* and *'pattern'* attributes respectively. The tensor concatenations show how the parameters W,V,b,c_X and c_Y etc. are intialized with the corresponding parameters of the attribute based CorrNets. (Best viewed in color)

The intuition behind this specific initialization is that the alignment between the dimensions capturing the three salient attributes have already been captured by the individual attribute based CorrNets. For example, forming the W matrix by a diagonal concatenation of the individual W_a, W_c and W_p matrices will ensure that the bag-of-apparel-words, bag-of-color-words and bag-of-pattern-words in the input interact only with their respective weight matrices W_a, W_c and W_p and no entanglement happens between the already aligned dimensions of each hidden representation. Similarly the V matrix and the encoding bias vector b are formed by concatenating the individual matrices obtained from the three attribute based CorrNets. The parameters during reconstruction time W' and V' are initialized similar to W and V in Equation 3. Figure 2 pictorially represents the initialization of the matrices W, V, b, c_X, c_Y for the model by concatenating or combining the individual weight and bias tensors obtained from the three pre-trained attribute based CorrNets.

However, this initialization only ensures that at the beginning of training (iteration 0) the hidden representations in the two modalities can be expressed as a concatenation of the hidden representations learned by each of the three attribute based CorrNet models, *i.e.*,

$$h_X(x) = [h_{X_a}, h_{X_c}, h_{X_p}], h_Y(y) = [h_{Y_a}, h_{Y_c}, h_{Y_p}]$$

Therefore, at the beginning of training each of the subvectors of $h_X(x)$ and $h_Y(y)$ are already aligned with each other. However, in order to explicitly guide the model further to adhere to this pre-trained alignment while updating the parameters in the training step, we add a regularizer $\Omega(W)$ to the loss term, designed as follows.

$$\Omega(W) = C_1\ell_2(W \circ \mathcal{D}) + C_2\ell_1(W \circ \mathcal{N}) \tag{5}$$

where $\mathcal{D}$ and $\mathcal{N}$ signify matrices comprising of the diagonal and the non-diagonal elements of W

$$\mathcal{D} = \begin{pmatrix} \mathbf{1}_{W_a} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{W_c} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}_{W_p} \end{pmatrix}, \qquad \mathcal{N} = \mathbf{1}_{\mathcal{D}} - \mathcal{D}$$

where $\mathbf{1}_M$ denotes a matrix of ones of the shape of M.

In equation (5), ℓ_1 and ℓ_2 denotes ℓ_1-norm and ℓ_2-norm respectively, $\circ$ denotes elementwise matrix multiplication, and C_1, C_2 are the corresponding regularization constants tuned as hyperparameters. This achieves the desired effect of a controlled way of updating the W and V matrices owing to the sparse-nature of the ℓ_1 normed regularization on the non-diagonal elements, in order to curb any "mixing" effect between the different alignments learnt earlier. However, the ℓ_2 norm applied on the diagonal elements does not apply any such sparsity constraints within each attribute, allowing the updates to capture other characteristics like the relative importance of each of the attributes etc.

Similar to the CorrNet setup, every training instance here is denoted by $o_i = (x_i, y_i)$, *i.e.*, a concatenation of the text and image representation, and one of the objectives of the model is to reconstruct o_i from each of the following hidden represenatations $h(x), h(y)$ or the concatenated hidden representation $h(o)$ using the appropriate decoding $z_X(\cdot), z_Y(\cdot)$ or the concatenation of $(z_X(\cdot), z_Y(\cdot))$ denoted by $z(\cdot)$.

In addition to the reconstruction loss, while the objective function in the original CorrNet maximizes the correlation between the hidden representations of the two modalities, we empirically found that instead, minimizing the squared euclidean distance between the representations performed better. The final objective function is given below, where $\mathcal{L}$ is the squared error loss and λ is the hyperparameter which balances the alignment between the hidden representation of the two modalities and the reconstruction losses over the N training instances.

$$\begin{aligned}\mathcal{J}_{\mathcal{Z}}(\theta) = \sum_{i=1}^{N} (&\mathcal{L}(o_i, z(h(o_i))) + \mathcal{L}(o_i, z_X(h(x_i))) \\ &+ \mathcal{L}(o_i, z_Y(h(y_i)))) + \lambda\mathcal{L}(h(X), h(Y))) + \Omega(W)\end{aligned} \tag{6}$$

5. Experiments

This section describes our data, the experimental setup of our proposed model motivating the use of the learnt common representations in various cross-modal applications.

5.1. Dataset

We have curated a large dataset (which would be released on acceptance of this work) of 0.72 million products (apparel, footwear and accessories for men and women) from five fashion retail sites. Each item generally has multiple product images, category of the product, its name and description, and optionally other attributes *e.g.* color, material, style-tip, etc.. We only use the product images, the product name, and gender information for all our experiments.

A training corpus of 0.67 million fashion items was constructed using four fashion sites and 47K products from the fifth site were completely sequestered away for testing purpose, for evaluating the generalization capability of our models. Further, from the training set, 20K products were taken out for validation purposes. The vocabulary for the catalog data was created from all unigrams appearing with a minimum frequency of 5 in the training corpus, and filtering through the *'apparel'*, *'color'* and *'pattern'* lexicons resulting in a total vocabulary of size 2K.

5.2. Evaluated Correlational Models

The following correlational models are evaluated on the different end-applications:

Baseline CorrNet Model: This is similar to the Baseline CorrNet Model (with euclidean distance based joint representation learning) described in Section 3

Attribute Based CorrNet Models: This corresponds to the three correlational networks (discussed in 4.1) trained on the three salient attributes *'apparel'*, *'color'* and *'pattern'*. Given a particular end application, the resulting ranked list of items (images or tags) of the three model are merged keeping the following domain requirements in mind i.e. when a particular fashion item is searched for, matching the *'apparel'* has a higher priority over its attributes (*'color'* or *'pattern'*). Hence the merge strategy applies the same principle and filters out all the results obtained from the *'color'* and *'pattern'* model which do not belong to the result-list of the *'apparel'* model, thus obtaining the final result-list as the concatenation of the following (where $\mathcal{R}$ denotes the ranked result list obtained from the corresponding model)

$$\mathcal{R} = [\mathcal{R}_{apparel} \cap \mathcal{R}_{color} \cap \mathcal{R}_{pattern}, \quad \mathcal{R}_{apparel} \cap \mathcal{R}_{color}, \\ \mathcal{R}_{apparel} \cap \mathcal{R}_{pattern}, \quad \mathcal{R}_{apparel}]$$

Align-and-Correlate CorrNet Model: This is the CorrNet model described in Section 4.2, which borrows the pretrained alignment from the attribute based CorrNet models as an initialization step and learns the hidden representation of the two modalities, combining the three salient attributes.

Though empirically it is seen for most of the end applications, the attribute based CorrNet model performs much better than the Align-and-Correlate CorrNet, the latter is a more principled architecture which is also easily usable by the downstream applications. Further, the attribute based CorrNet model needs a post-processing step which requires additional knowledge of the priority in the domain (for e.g. matching the *'apparel'* has a higher priority over other attributes), whereas the Align-and-Correlate CorrNet tries to learn the priority order.

Hyper-parameter tuning: The model hyper-parameters were tuned over the following set: hidden layer dimension $\in \{256, 512\}$, learning rate $\in \{$1e-2,1e-3,1e-4$\}$, λ associated with the correlation term $\in \{0.1, 0.5, 2, 5, 10\}$, regularization constant $\in \{$1e-3, 1e-4$\}$. The optimization methods tried were stochastic gradient descent, adaptive learning rate method (adadelta)[33], and rmsprop[28]. The hyperparameter selection was done over the validation set.

5.3. Annoy index

Using the 256 dimensional representation we create two gender-specific indices (for *'men'* and *'women'*) over the product images in the test catalog for fast retrieval of relevant images, given a query or image. For fast approximate nearest neighbour search required in each of the end applications, we use the Annoy (Approximate Nearest Neighbors Oh Yeah `https://github.com/spotify/annoy`) package.

5.4. Application 1: Cross-modal Image Retrieval

In this setting, the user provides a natural language query and the retrieval system is required to fetch a ranked list of relevant images that match the query description. We construct the test query set of 3780 unique product names [1] from the test catalog. A few example outputs of the proposed method are shown in Table 1 and 2 of the supplementary material.

Evaluation metrics: We present the automated evaluation for the cross modal image retrieval setting and report the standard information retrieval metrics, namely, Mean Average Precision (MAP), Mean Reciprocal Rank (MRR) and Precision@Top-K over the top 10 retrieved results. For the automated evaluation, since a gold standard comprising of relevant results for a query is not available, we use the following notion to decide whether a retrieved product is a right match or not. If the product name of the retrieved image covers at least a threshold fraction of θ of the given query words, we consider that product to be a correct result for the query. By varying this threshold θ from 0.1 to 0.9, we obtain the MAP and MRR values for our method and each of the competing methods. Further for a reasonable value of this threshold ($\theta = 0.6$), we also show a plot of the precision over the top-K retrieved results, by varying K from 1 to 10. We compare our proposed approach with the

[1] The nature of these queries follow the template '*gender apparel-attributes apparel*' (for *e.g.*. 'men red and black checkered formal shirt'), i.e each query word belonging to a *'type'* namely, gender-type (*male, female*) or attribute-type or apparel type(*crop top, formal shirt*). Further every attribute-type have different sub-types e.g. color-type (*yellow, pink*) , pattern-type (*striped, polka-dotted*)

following skyline:

Index based text retrieval with noisy query: One solution for handling this kind of cross-modal application is to use an image tagging system which tags every catalog image and given the natural language query, retrieves relevant images for it. To simulate a skyline version of the same, we consider the catalog information (*i.e.*, the gold data pertaining to the title and description of the product) to be available and construct a standard Lucene (`https://lucene.apache.org/`) index out of this. We use this Lucene index to retrieve a list of relevant items [2] for the query. Since in reality, such automatic taggers cannot be expected to give 100% accuracy, we then design impoverished versions of such an automated image tagging system having $x\%$ accuracy by randomly replacing on an average $(100 - x)\%$ words in the given query by other words in the same type, for e.g. transforming *'blue jacket'* query to *'yellow jacket'* or *'blue shirt'*. Since it is difficult to implement the actual image tagging algorithm we have taken this simulated approach, where we synthetically emulate such a automated image tagging system having 70% accuracy and compare with the proposed methods and other baselines.

Figure 3(a) and (b) shows MAP and MRR against the threshold of query coverage in the retrieved image names, thus studying the performance of the system as the evaluation becomes stricter. The index based text retrieval system indicates the best performance that can be possibly achieved and serves as a skyline. The baseline systems which emulate an automated image tagging system having 30% noisy tags performs better than our proposed method, as it is allowed to query on the lucene index based on its detected tags. However, on further impoverishing the index by missing tags causes a sharp performance degradation, making it perform significantly worse than our models. Comparison with these baselines and skylines establishes that all of these methods require data in both modalities, in order to function properly. Figure 3(c) shows that the Precision@top-K over the top-10 retrieved results, where the threshold of the query coverage is fixed at a reasonable fraction of 0.6. Further, comparing between the three variations of the CorrNet model, it is clear that the Align-and-Correlate CorrNet far surpasses (showing around 5-20% improvement) the baseline in performance in terms of each of these metrics. Though the attribute based CorrNet in-turn performs much better (with around 10% improvement) than the Align-and-Correlate model, the latter is more a principled approach and does not require any domain based post-processing step.

[2] On firing a phrase query, lucene searches for contiguous occurrence of the exact phrase, allowing an edit distance of up to 2 moves. On failing, it backs-off to more relaxed searches requiring atleast all the query words to appear in the catalog description

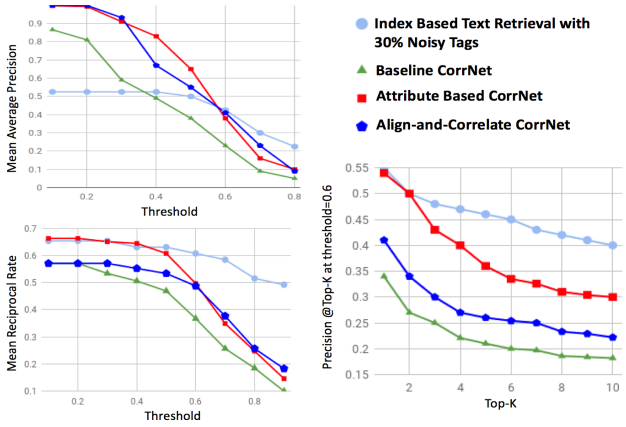

Figure 3. (a),(b) are the Mean Average Precision(MAP), Mean Reciprocal Rank(MRR) plotted against the threshold of query coverage, shown for different methods for the top 10 retrieved images for a given text query. (c) shows the Precision@top-K over varying K between 1 to 10, at a constant query coverage threshold of 0.6

5.5. Application 2: Visual Search

The other setting where our model can be evaluated is the visual search application where the system has to retrieve similar looking images, for a given image query. One of the widely used state-of-the-art visual search systems is based on the *fc6*-layer representation learnt by a Convolutional Neural Network(CNN) [15] which have been popularly used for various image understanding tasks [21, 20]. By taking this *fc6*-layer representation we map every image to a 4096-dimensional vector representation and further take the euclidean distance in this space to retrieve similar images for given query image. At test time, given a query image, each of these competing systems use their respective nearest neighbor indices to come up with a ranked list of similar images. Once again, since the gold data of the 'correct' matching images for a target image is not available, we use the available catalog name of the corresponding image, as a proxy of its relevance to the query. Similar to the notion of query overlap employed previously, here we consider a retrieved image to be a correct match for the target, if the product name of the query contains at least a threshold of θ fraction of the words in the title of the target product. By varying this threshold fraction of coverage from 0.1 to 0.9 we obtain a plot of the performance of the system as the evaluation gets stricter.

Figure 4(a)-(c) compares the performance of the contemporary state-of-the-art BVLC CaffeNet and the different proposed variations of CorrNet, using the metrics MAP, MRR, and Precision@Top-K used earlier. It can be seen from the graphs that the CorrNet models beat the standard visual search in all the three metrics, even by using a much lower representation size. Further, the Align-and-Correlate CorrNet model performs (around 5%) better than the baseline CorrNet in performance in terms of all three metrics,

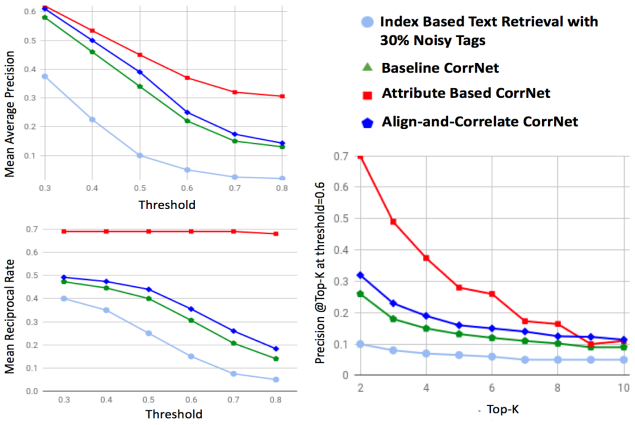

Figure 4. (a),(b) plot the MAP and MRR of the different methods for the top 10 retrieved images for a given image and (c) shows the Precision@top-K by varying K at a fixed threshold of 0.6

while the attribute based CorrNet beats both of the variants by a large margin (with over 10-20% improvement). A few example outputs of the proposed method for this setting are given in Table 3 in the supplementary material.

men's fashion	**blue-printed slim-fit casual-shirt**	**beige chino trousers**	**black axis dp running shoes**
	light-blue printed linen slimfit casual-shirt	khaki cambridge-fit-chinos	black axis running shoes
	grey-light blue printed casual shirt	dark beige chino trousers	black axis v3dp running shoes
	purple slim fit casual shirt	off-white cambridge fit chinos	black fashion running shoes
	teal blue custom fit casual shirt	brown chino trousers	black ignite mesh running shoes
	blue-green checked slimfit casual shirt	khaki stretch chino trousers	black synthesis running shoes
women's fashion	**black & green dot print top**	**multicolor wide-leg trousers**	**black shift dress**
	blue black dot print top	orange-printed palazzo	black blouson dress
	black white geometric print top	maroon palazzo trousers	blue-black lace shift dress
	black pink polka dot print top	purple trousers	black sheath dress
	black white abstract print top	navy trousers	black bodycon dress
	maroon black polka dot print top	purple harem pants	black polka dotted shift dress
		orange-khaki trousers	black gold bodycon dress

Table 1. Related product names suggested by our proposed Align-and-Correlate CorrNet model, given a product name.

5.6. Application 3: Automated Image Tagging

We are now interested in getting a ranked list of product names or tags that are suitable for a given image. The evaluation is done over the top-20 product-names retrieved for each of the 47K queries. As a strict match of the closeness of the suggested product names to the true title of the product, we consider Jaccard (`https://en.wikipedia.org/wiki/Jaccard_index`) similarity between the two phrases. Further, as a relaxed measure, we consider the fraction of words in the true product title that appear in one of the suggested product names. Figure 5 shows the plot of these two best matching scores over the top-K retrieved results by varying K from 1 to 20, for the three CorrNet variants. As seen from the plot, the Align-and-Correlate CorrNet performs best of the three, and gives around 5% boost in performance over the baseline CorrNet. A few example outputs of the proposed method for this setting are given in Table 4 of the supplementary material.

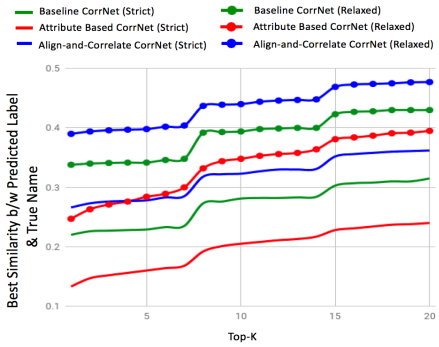

Figure 5. Best strict (Jaccard Similarity) and relaxed (Fractional Overlap) match obtained over the top-K retrieved tags, plotted against K for image tagging.

5.7. Application 4: Related Phrase Suggestion

Finally, we consider the case where the user provides an input phrase and the system returns the nearest neighboring words or product names. The input phrase can be a single word or tag (*e.g. 'jacket'*) or a complete product name (usually of the template '*gender apparel-attribute(s) apparel*'). The nearest neighbors are essentially the product names whose common representation has the shortest euclidean distance from the representation of the input phrase. Table 1 shows the most closely related product names for a given target product. As exemplified in this table, our method suggests different variations of the product e.g. brand variations (*e.g. 'black victor brogues'* for *'loafers'*) or type-variations (*e.g. 'blouson dress'* or *'sheath dress'* for *'dress'*) or different attributes variations (*e.g.* 'cream colored t-shirt' or 'coral red t-shirt' for 'peach-colored t-shirt') or similar-looking category of apparels (*e.g. 'palazzo trousers'* or *'harem pants'* for *'wide-legged trousers'*). More examples are provided in Table 5 and 6 of the supplementary.

6. Conclusions and Extensions

In this paper we extensively evaluated the feasibility of using joint multimodal representations for searching for apparel and accessories on fashion e-commerce portals. These representations were learnt over a large curated fashion dataset of over 700K images gathered from multiple fashion e-commerce portals. We also proposed novel variations of correlational autoencoder based model for learning joint multimodal representations, namely, 'Attribute based CorrNet' and 'Align-and-Correlate CorrNet', which were empirically shown to give 10-20% boost in precision over the standard CorrNet, for various multimodal end applications like cross-modal image retrieval, visual search, image tagging, and query expansion. As an extension of this work, we aim to solve more complex forms of multimodal retrieval which support AND and OR style queries by enabling such boolean compositions in representation learning models.

References

[1] S. Akaho. A kernel method for canonical correlation analysis. *In Proc. Int'l Meeting on Psychometric Society*, 2001.

[2] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool. Apparel classification with style. In *11th Asian Conference on Computer Vision (ACCV)*, 2012.

[3] S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran. Correlational neural networks. *Neural computation*, 2016.

[4] S. Chandar, S. Lauly, H. Larochelle, M. M. Khapra, B. Ravindran, V. C. Raykar, and A. Saha. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1853–1861, 2014.

[5] H. Chen, A. Gallagher, and B. Girod. Describing clothing by semantic attributes. In *ECCV*, 2012.

[6] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan. Deep domain adaptation for describing people based on fine-grained clothing attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[7] R. Cruz-Cano and M.-L. T. Lee. Fast regularized canonical correlation analysis. *Computational Statistics & Data Analysis*, 2014.

[8] S. Gella, R. Sennrich, F. Keller, and M. Lapata. Image pivoting for learning multilingual multimodal representations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2829–2835, 2017.

[9] A. Habibian, T. Mensink, and C. G. Snoek. Videostory: A new multimedia embedding for few-example recognition and translation of events. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 17–26, New York, NY, USA, 2014. ACM.

[10] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28, 1936.

[11] W. Hsieh. Nonlinear canonical correlation analysis by neural networks. *Neural Networks*, 13(10), 2000.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[13] Y. Jing, D. Liu, D. Kislyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel. Visual search at pinterest. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[14] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg. Where to buy it: Matching street clothing photos in online shops. In *International Conference on Computer Vision (ICCV)*, 2015.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, 2012.

[16] S. I. C. K. C. G. S. Mikolov, T. and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 2013.

[17] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 689–696, 2011.

[18] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 689–696, 2011.

[19] F. Å. Nielsen, L. K. Hansen, and S. C. Strother. Canonical ridge analysis with ridge parameter optimization, may 1998.

[20] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 14, Washington, DC, USA, 2014. IEEE Computer Society.

[21] M. Ozeki and T. Okatani. Understanding convolutional neural networks in terms of category-level attributes. In *Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II*, 2014.

[22] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[23] J. R, M. M. Khapra, S. Chandar, and B. Ravindran. Bridge correlational neural networks for multilingual multimodal representation learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 11–20, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[24] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 251–260, New York, NY, USA, 2010. ACM.

[25] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *TACL*, 2, 2014.

[26] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep boltzmann machines. *Journal of Machine Learning Research*, 15:2949–2980, 2014.

[27] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep boltzmann machines. *Journal of Machine Learning Research*, 15, 2014.

[28] T. Tieleman and G. Hinton. RMSprop Gradient Optimization.

[29] H. Vinod. Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 4(2):147 – 166, 1976.

[30] W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. In *ICML*, 2015.

[31] W. Wang, B. C. Ooi, X. Yang, D. Zhang, and Y. Zhuang. Effective multi-modal retrieval based on stacked auto-encoders. *Proc. VLDB Endow.*, 7(8):649–660, Apr. 2014.

[32] K. Yamaguchi, H. Kiapour, and T. L. Berg. Paper doll parsing: Retrieving similar styles to parse clothing. In *International Conference on Computer Vision (ICCV)*, 2013.

[33] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[34] X. Zhai, Y. Peng, and J. Xiao. Learning cross-media joint representation with sparse and semisupervised regularization. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(6):965–978, June 2014.

[35] Y. Zheng, Y. Zhang, and H. Larochelle. A deep and autoregressive approach for topic modeling of multimodal data. *CoRR*, abs/1409.3970, 2014.

[36] Y. Zheng, Y. Zhang, and H. Larochelle. Topic modeling of multimodal data: An autoregressive approach. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1370–1377, 2014.