OutfitTransformer: Outfit Representations for Fashion Recommendation

Rohan Sarkar[1,2], Navaneeth Bodla[2], Mariya Vasileva[2], Yen-Liang Lin[2], Anurag Beniwal[2], Alan Lu[2], and Gerard Medioni[2]

[1]Purdue University, West Lafayette, [2]Amazon

Abstract

Predicting outfit compatibility and retrieving complementary items are critical components for a fashion recommendation system. We present a scalable framework, OutfitTransformer, that learns compatibility of the entire outfit and supports large-scale complementary item retrieval. We model outfits as an unordered set of items and leverage self-attention mechanism to learn the relationships between items. We train the framework using a proposed set-wise outfit ranking loss to generate a target item embedding given an outfit, and a target item specification. The generated target item embedding is then used to retrieve compatible items that match the outfit. Experimental results demonstrate that our approach outperforms state-of-the-art methods on compatibility prediction, fill-in-the-blank, and complementary item retrieval tasks.

1. Introduction

There are two main tasks for a fashion outfit recommendation system: outfit *compatibility prediction* (CP) and large-scale *complementary item retrieval* (CIR). For CP, the task is to determine whether a set of fashion items in an outfit go well together. For CIR, the task is to complete a partial outfit by finding a compatible item from a large database. Figure 1 illustrates our proposed method. Given an outfit, we want to predict how well its constituent items go together. Also, given a partial outfit (such as a bag, shoes, and pants) and a target item description (*e.g.*, "top"), we want to retrieve compatible items to complete the outfit.

Prior work such as [15, 17, 19–21] addresses the pairwise item-level compatibility problem and achieves state-of-the-art results but does not explicitly model outfit-level compatibility. Some methods optimize for compatibility at an outfit-level [3, 4, 6–8]. However, these approaches are mainly designed for classification tasks – CP and fill-in-the-blank (FITB) – but they do not address the large-scale CIR task. CSA-Net [12] proposes a method for large-scale CIR, but it does not learn an outfit-level representation that can explicitly capture compatibility of a target item to the outfit as a whole. It searches compatible items for each item in the outfit at a paired-category level (*e.g.*, top to shoe, bottom to shoe) and fuses the ranking scores for different query items to obtain the final rankings. Lorbert *et al.* [13] use a single layer self-attention based framework for outfit generation, but do not explicitly model compatibility. Instead, our idea is to learn an outfit-level representation for both compatibility prediction and large-scale retrieval of complementary items. Using outfit-level representations can more effectively capture complex feature correlations among multiple items in the outfit, as opposed to considering pairs of items at a time. Further, our method has a much smaller indexing size than [12] which is important for practical applications (cf. Section 2.2(b)).

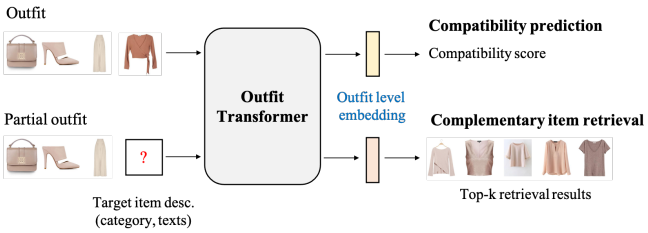
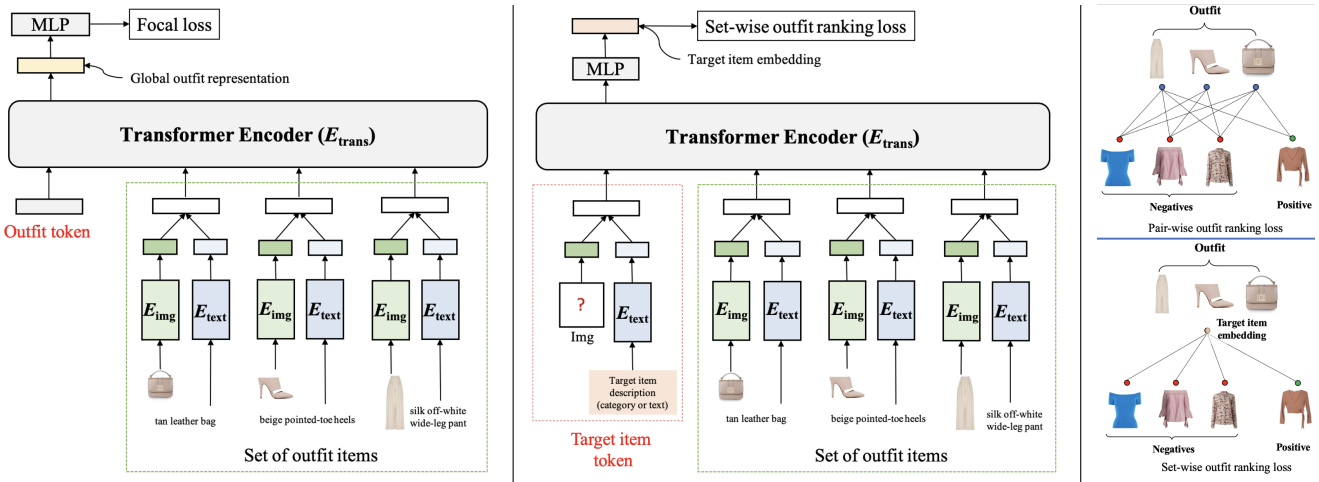


Figure 1. OutfitTransformer learns an outfit-level representation for a set of outfit items to address the CP and CIR tasks. For CIR, it learns a single embedding encoding overall compatibility of the partial outfit, and a target item description that is used to retrieve compatible items cohesively matching the entire outfit using KNN search. For CP, an outfit-level representation capturing overall outfit compatibility is learnt to predict a compatibility score.

Our framework, OutfitTransformer, is based on a transformer encoder architecture. Attention mechanisms [3, 12, 13, 16] have also been used in fashion recommendation systems. [12, 16] use attention to understand complementary relationships in a pairwise manner. The self-



(a) Compatibility prediction (CP) (b) Complementary item retrieval (CIR) (c) Loss comparison

Figure 2. System overview of our framework for CP and CIR. We model outfits as an unordered set of items. We use an image encoder (E_{img}) and a text encoder (E_{text}) to extract the image and text features. (a) For CP, we train the transformer encoder using a focal loss [11] to learn a global outfit representation to predict an outfit compatibility score. (b) For CIR, given an outfit and a target item description, we train the transformer encoder to learn a target item embedding that can be used for retrieving compatible items to complete an outfit. We train the framework using the proposed set-wise outfit ranking loss in an end-to-end manner (details in Section 2.2(a)). Comparison of pair-wise outfit ranking loss [12] and our proposed set-wise outfit ranking loss is shown in (c).

attention mechanism in the transformer captures the higher-order relationships [9] between the outfit items beyond pair-wise [12, 15, 17], and does not require ordered items as inputs. For CP, we train the OutfitTransformer with a classification loss and design an outfit token to capture a global outfit representation that encodes the compatibility relationships among all the items in the outfit (Fig. 2(a)). For CIR, we design a target item token that encodes both the compatibility of the partial outfit and a target item description to generate the embedding of the target item. This embedding is used to retrieve compatible items from a database (Fig. 2(b)). We train our framework using a proposed set-wise outfit ranking loss that encourages compatible items to be embedded closer to the overall representation of a set of outfit items. Our design allows extraction of a single target item embedding enabling large-scale indexing and retrieval.

We evaluate our method on the public Polyvore Outfits dataset [17]. The experimental results show that our approach outperforms state-of-the-art techniques in compatibility prediction, fill-in-the-blank (FITB), and complementary item retrieval tasks (cf. Section 3).

2. Proposed Approach

2.1. Fashion Outfit Compatibility Prediction

The CP task predicts the compatibility of all the items in an outfit. Given an outfit $O = \{(\mathrm{I}_i, \mathrm{T}_i)\}_{i=1}^{L}$, where I_i is the image, T_i is the corresponding text description for an item i, we learn a non-linear function that predicts a compatibility score in $[0, 1]$, where 1 indicates perfect compatibility.

As shown in Fig. 2(a), the item images and their text descriptions are fed into an image (E_{img}) and text encoder (E_{text}) respectively to extract the image and text feature vectors (see Section 2.3 for the details). We concatenate the extracted image and text feature vectors to generate an item feature vector $u_i = E_{\text{img}}(\mathrm{I}_i) \parallel E_{\text{text}}(\mathrm{T}_i)$, where $\parallel$ denotes a concatenation operation. The feature vectors of all items in an outfit are represented by the set $F = \{u_i\}_{i=1}^{L}$.

We introduce the outfit token whose state at the output of the transformer encoder serves as the global outfit representation capturing compatibility relationships between items in the outfit using the self-attention mechanism. We model outfits as an unordered set of items as the overall outfit compatibility is invariant to the order of the items. Thus, positional encodings used in NLP [18] and vision transformers [5] are not required in our case.

The outfit token ($\mathrm{x}_{\text{Outfit}}$) is a learnable embedding that is prepended to the set of outfit feature vectors F and fed into the transformer encoder E_{trans}. The state of the outfit token at the output of the transformer encoder serves as the global outfit representation which is subsequently fed into the MLP that predicts an overall outfit compatibility score:

$$\mathrm{c} = \mathrm{MLP}\left(E_{\text{trans}}\left(\mathrm{x}_{\text{Outfit}}, F\right)\right) \tag{1}$$

Our framework ($E_{\text{trans}}, E_{\text{img}}, E_{\text{text}}$) is trained in an end-to-end manner using focal loss [11].

2.2. Complementary Item Retrieval

The CIR task is to retrieve an item that is both compatible with the partial outfit and matches a specified item description to complete the outfit. Specifically, given a set of partial outfit items and a target item specification, the goal is to generate a target item embedding that can be used to retrieve compatible items.

We introduce the target item token whose state at the output of the transformer encoder serves as the target item representation that explicitly takes into consideration both compatibility with the partial outfit, and the target item description. The target item token s (cf. Fig. 2 (b)) includes an item description T for the target item that we want to retrieve, and an empty image represented by $\mathrm{x}_{\mathrm{Img}}$. The target item token is defined as: $\mathrm{s} = \mathrm{x}_{\mathrm{Img}} \parallel E_{\mathrm{text}}(\mathrm{T})$. The intuition behind designing the target item token in this manner is that, during inference, the target image is unknown but users can provide a description for the item they are searching for. We simulate a similar setting when training the framework for the retrieval task. Our framework is generic and the target item description can be provided in different forms such as category, text description, tags, etc.

The transformer encoder takes as input the set of feature vectors F of the partial outfit, and the target item token s, which is subsequently fed into a MLP that generates the target item embedding.

$$\mathrm{t} = \mathrm{MLP}\left(E_{\mathrm{trans}}(\mathrm{s}, F)\right) \tag{2}$$

To learn this target item embedding, we first pre-train our framework on the CP task and then train the model with the proposed set-wise outfit ranking loss discussed next. Pre-training allows E_{img} and E_{text} to capture fashion-specific features and E_{trans} to capture outfit compatibility relationships which improves CIR performance significantly.

(a) Set-wise Outfit Ranking Loss: Previous approaches [15, 17] use a triplet loss to learn relationships only between a pair of items but do not consider the relationship between all items in the outfit. To address this, the outfit ranking loss [12] is proposed which considers the pairwise compatibility of target items with all the items in the outfit, as shown in the top of Fig. 2 (c). In contrast, our approach generates a single target item embedding t that already captures the compatibility relationships for a set of outfit items and hence does not require pairwise comparisons with individual outfit items, as shown in the bottom of Fig. 2 (c). Thus we can directly train our set-wise ranking loss using triplets which reduces the complexity of [12] from $\mathcal{O}(LS)$ to $\mathcal{O}(S)$ during training, where L denotes the outfit length and S denotes the number of positive and negative samples.

Given an outfit we randomly pick an item as positive and the remaining items as the partial outfit. We propose a curriculum learning approach specifically designed for CIR by hierarchically sampling the negatives first from the same high-level category as the positive item and subsequently sampling harder negatives from more fine-grained categories. The set-wise outfit ranking loss is designed to optimize the relative distances between samples such that the target item embedding moves closer to the positive embedding and farther apart from the negative embeddings.

(b) Indexing and retrieval for complementary items: Not all the methods for CP can support indexing for retrieval [15]. Our design allows extraction of individual item feature vectors during indexing and generate a single item embedding during inference. Specifically, during indexing, we use the trained image and text encoder to extract the item features. During inference, given the partial outfit and a target item description, our framework generates a single target item embedding, which is then used to search for compatible items from the database using off-the-shelf KNN search tools (*e.g.*, [1, 2]) which makes the search efficient even for a large database (*e.g.*, with millions of items).

Our framework offers two advantages as compared to prior works. First, we require smaller indexing size compared to previous approaches that use subspace embeddings. For indexing, Type-aware [17] and CSA-Net [12] generate multiple embeddings of each item for each of the target categories and therefore the indexing size grows linearly with the number of categories. Because we are not learning subspaces, our approach is independent of the number of categories. Second, in [12], for each item in the outfit, a target category-specific embedding is extracted, which is used to retrieve compatible items from the database. This has to be repeated exhaustively for each item in the query outfit. In contrast, in our framework, retrieval can be performed in a single step regardless of outfit length.

2.3. Implementation Details

The image encoder uses a ResNet-18 initialized with ImageNet pre-trained weights. The text encoder uses a pre-trained SentenceBERT [14], on top of which we add a fc

Method	Features	PO-D	PO
BiLSTM + VSE [6]	ResNet-18 + Text	0.62	0.65
GCN (k=0) [10]	ResNet-18	0.67	0.68
SiameseNet [17]	ResNet-18	0.81	0.81
Type-Aware [17]	ResNet-18 + Text	0.84	0.86
SCE-Net [15]	ResNet-18 + Text	-	0.91
CSA-Net [12]	ResNet-18	0.87	0.91
OutfitTransformer (Ours)	ResNet-18	0.87	0.92
OutfitTransformer (Ours)	ResNet-18 + Text	**0.88**	**0.93**

Table 1. Comparison of our model with state-of-the-art methods on the CP task using the AUC metric [6].

Method	Polyvore Outfits-D				Polyvore Outfits			
	FITB	R@10	R@30	R@50	FITB	R@10	R@30	R@50
Type-Aware [17]	55.65	3.66	8.26	11.98	57.83	3.50	8.56	12.66
SCE-Net Average [15]	53.67	4.41	9.85	13.87	59.07	5.10	11.20	15.93
CSA-Net [12]	59.26	5.93	**12.31**	**17.85**	63.73	8.27	15.67	20.91
OutfitTransformer (Ours)	**59.48**	**6.53**	12.12	16.64	**67.10**	**9.58**	**17.96**	**21.98**

Table 2. Comparison of our model with state-of-the-art methods on the FITB (using accuracy) and CIR tasks (using recall@top-k).

layer. During training, we finetune the weights of the image encoder and the fc layer of the text encoder. We extract a 64-dimensional image and a 64-dimensional text embedding and concatenate them to generate 128-dimensional item embeddings before feeding them into the transformer encoder. We use a six-layer transformer encoder with 16 heads. For the retrieval task, we set the margin m for the set-wise outfit ranking loss as 2 and sample 10 negatives for each outfit. We use a batch size of 50 and optimize using ADAM with an initial learning rate of $1e-5$ and reducing the learning rate by half in steps of 10.

3. Experiments

We evaluate our method on Polyvore-Outfits [17] nondisjoint and disjoint datasets (cf. Tables 1, 2) and compare performance with state-of-the-art baselines [4,6,12,15, 17,19] on three different tasks: (1) *Outfit Compatibility Prediction (CP) task* that predicts the compatibility of items in an outfit. (2) *Fill in the Blank (FITB) task* that selects the most compatible item for an incomplete outfit given a set of candidate choices (*e.g.*, 4 candidates). (3) *Outfit Complementary Item Retrieval (CIR) task* that retrieves complementary items from the database for a target category given an incomplete outfit.

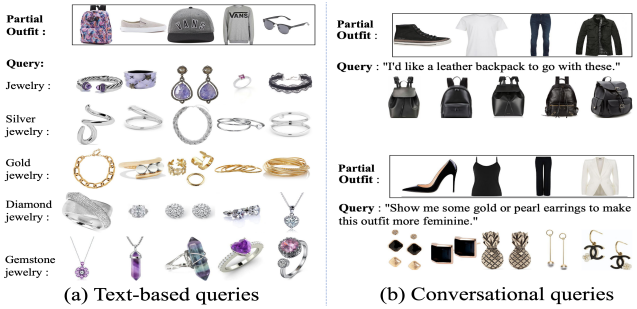
CP: We compare the CP performance with the state-of-the-art methods in Table 1 by using the standard metric AUC [6]. We observe that using just image features; we outperform other methods that use both image and text features on the CP task. Using text features boosts performance further. [12, 15, 17] employ a pairwise model requiring careful selection of negatives and data augmentation. Our approach uses the outfit compatibility data provided without using any additional strategies and still outperforms the state of the art methods. From Table 1, we observe that transformers can learn better compatibility relationships than other methods [4, 6] that learn compatibility at an outfit-level.

FITB and CIR: Lobert et al. [13] propose to use pre-trained ImageNet embeddings and category for retrieval using self-attention. For evaluation, we adopt their strategy using our own implementation using a transformer and observe that their FITB accuracy on the Polyvore Outfits dataset is 41.61%. We investigate several strategies such as pre-training on the CP task, curriculum learning, hard-negative mining and observe an improvement in FITB performance of 13.14%, 3.77% and 2.62% respectively. Our method yields a FITB performance of 58.92% when using images and category and 67.10 % using images and text.

For retrieval, we use the same testing setup as CSA-Net [12], and compare the performance of our method with state of the art methods CSA-Net [12], Type-aware [17] and SCE-Net average [15]. For evaluation, we use the category as our target item description for retrieving complementary items and use recall@top-k metric that measures the rank of the ground-truth item similar to [12]. From Table 2, we observe that we outperform all the methods on the non-disjoint dataset. On the disjoint dataset, our performance on recall@top-10 is better than CSA-Net, but slightly worse on recall@top-30 and recall@top-50. We conjecture the reason for the performance drop might be because there are fewer outfits on the disjoint set, and transformers typically require large training data to generalize well. Also, the authors in CSA-Net discuss that the rank of the ground truth is not a perfect measure for evaluating retrieval performance. Some example retrieval results are shown in Fig. 3.

4. Conclusion

Our model learns outfit-level representations and outperforms state-of-the-art approaches on the Polyvore Outfits dataset in three established tasks: CP, FITB, and CIR.



(a) Text-based queries (b) Conversational queries

Figure 3. For each partial outfit and a text-based query this figure shows the top 5 retrieved items that are both compatible with the outfit, and match the text query.

References

[1] https://github.com/nmslib/hnswlib. 3

[2] https://github.com/facebookresearch/faiss. 3

[3] Wen Feng Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. Pog: Personalized outfit generation for fashion recommendation at alibaba ifashion. *SIGKDD*, 2019. 1

[4] Guillem Cucurull, Perouz Taslakian, and David Vazquez. Context-aware visual compatibility prediction. In *CVPR*, 2019. 1, 4

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[6] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S. Davis. Learning fashion compatibility with bidirectional lstms. In *ACM MM*, 2017. 1, 3, 4

[7] Wei-Lin Hsiao and Kristen Grauman. Learning the latent "look": Unsupervised discovery of a style-coherent embedding from fashion images. In *ICCV*, 2017. 1

[8] Wei-Lin Hsiao and Kristen Grauman. Creating capsule wardrobes from fashion images. In *CVPR*, 2018. 1

[9] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer. *ArXiv*, 2018. 2

[10] Kedan Li, Chen Liu, and David Forsyth. Coherent and controllable outfit generation, 2019. 3

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *TPAMI*, 2020. 2

[12] Yen-Liang Lin, S. Tran, and Larry Davis. Fashion outfit complementary item retrieval. In *CVPR*, 2020. 1, 2, 3, 4

[13] Alexander Lorbert, David Neiman, Arik Poznanski, Eduard Oks, and Larry Davis. Scalable and explainable outfit generation. In *CVPR Workshop*, 2021. 1, 4

[14] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019. 3

[15] Reuben Tan, Mariya I. Vasileva, Kate Saenko, and Bryan A. Plummer. Learning similarity conditions without explicit supervision. In *ICCV*, 2019. 1, 2, 3, 4

[16] Meet Taraviya, Anurag Beniwal, Yen-Liang Lin, , and Larry Davis. Personalized compatibility metric learning. *KDD Workshop*, 2021. 1

[17] Mariya I. Vasileva, Bryan A. Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *ICCV*, 2018. 1, 2, 3, 4

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2

[19] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *CVPR*, 2017. 1, 4

[20] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *ICCV*, 2015. 1

[21] X. Yang, Yunshan Ma, Lizi Liao, M. Wang, and Tat-Seng Chua. Transnfcm: Translation-based neural fashion compatibility modeling. *ArXiv*, 2019. 1