Attention-based Fusion for Outfit Recommendation

Katrien Laenen
katrien.laenen@kuleuven.be
Department of Computer Science, KU Leuven, Belgium

Marie-Francine Moens
sien.moens@kuleuven.be
Department of Computer Science, KU Leuven, Belgium

Figure 1: Example outfit in the Polyvore68K dataset. Fine details, such as the heels of the sandals, the flower applique on the dress and the red pendants of the bracelet, determine that these items match nicely. These details should therefore be captured in the item representations.

ABSTRACT

This paper describes an attention-based fusion method for outfit recommendation which fuses the information in the product image and description to capture the most important, fine-grained product features into the item representation. We experiment with different kinds of attention mechanisms and demonstrate that the attention-based fusion improves item understanding. We outperform state-of-the-art outfit recommendation results on three benchmark datasets.

CCS CONCEPTS

• **Information systems → Online shopping**; • **Computing methodologies → Image representations**; **Matching**; Natural language processing.

KEYWORDS

outfit recommendation, item understanding, attention, attention-based fusion

ACM Reference Format:
Katrien Laenen and Marie-Francine Moens. 2019. Attention-based Fusion for Outfit Recommendation. In *Proceedings of Workshop on Recommender Systems in Fashion, 13th ACM Conference on Recommender Systems (recsysXfashion'19).* ACM, New York, NY, USA, 6 pages.

recsysXfashion'19, September 20, 2019, Copenhagen, Denmark

1 INTRODUCTION

With the explosive growth of e-commerce content on the Web, recommendation systems are essential to overcome consumer overchoice and to improve user experience. Often users shop online to buy a full outfit or to buy items matching other items in their closet. Webshops currently only offer limited support for these kinds of searches. Some webshops offer a *people also bought* feature as suggestions for compatible clothing items. However, items that are bought together by others are not necessarily compatible with each other, nor do they necessarily correspond with the taste and style of the current user. Another feature some webshops provide is *shop the look*. This enables to buy all clothing items worn together with the viewed item in an outfit which is usually put together by a fashion stylist. However, this scenario does not provide alternatives that might appeal more to the user.

In this work, we tackle the problem of outfit recommendation. The goal of this task is to compose a fashionable outfit either from scratch or starting from an incomplete set of items. Outfit recommendation has two main challenges. The first is *item understanding*. Fine details in the garments can be important for making combinations. For example, the items in Figure 1 match nicely because of the red heels of the sandals, the red flowers on the dress and the red pendants of the bracelet. These fine-grained product details should be captured in the item representations. Moreover, usually there is also a short text description associated with the product image. These descriptions point out certain product features and contain information which is useful for making combinations as well. Hence, there is a need to effectively integrate the visual and textual item information into the item representations. The second challenge in outfit recommendation is *item matching*. Item

compatibility is a complex relation. For instance, assume items A and B are both compatible with item C. In that case items A and B can be, but are not necessarily, visually similar. Moreover, items A and B can be, but are not necessarily, also compatible with each other. Furthermore, different product features can play a role in determining compatibility depending on the types of items being matched, as illustrated in [11].

This work will focus on *item understanding*. Our outfit recommendation system operates on region-level and word-level representations to bring product features which are important to make item combinations to the forefront as needed. The contributions of our work are threefold. Firstly, our approach works on a finer level of image regions and words. In contrast, previous approaches to outfit recommendation work on a more coarse level of full images and sentences. Secondly, we explore different attention mechanisms and propose an attention-based fusion method which fuses the visual and textual information to capture the most relevant product features into the item representations. Attention mechanisms have not yet been explored in outfit recommendation systems to improve item understanding. Thirdly, we improve state-of-the-art outfit recommendation results on three datasets.

The remainder of this paper is structured as follows. In Section 2 we review other works on outfit recommendation. Then, Section 3 describes our model architecture. Next, Section 4 contains our experimental setup. The results of the conducted experiments are analysed in Section 5. Finally, Section 6 provides our conclusions and directions for future work.

2 RELATED WORK

The task of outfit fashionability prediction requires to uncover which items go well together based on item style, color and shape. This can be learned from visual data, language data or a combination of the two. Currently, two approaches are common to tackle outfit fashionability prediction. The first one is to infer a feature space where visually compatible clothing items are close together. [13] use a Siamese convolutional neural network (CNN) architecture to infer a compatibility space of clothing items. Instead of only one feature space, multiple feature spaces can also be learned to focus on certain compatibility relationships. [3] propose to learn a compatibility space for different types of relatedness (e.g., color, texture, brand) and weight these spaces according to their relevance for a particular pair of items. [11] infer a compatibility space for each pair of item types (i.e., tops and bottoms, tops and handbags) and demonstrate that the embeddings specialize to features that dominate the compatibility relationship for that pair of types. Moreover, their approach also uses the textual descriptions of items to further improve the results. The second common approach to outfit fashionability prediction is to obtain outfit representations and to train a fashionability predictor on these outfit representations. In [10] a conditional random field scores the fashionability of a picture of a person's outfit based on a bag-of-words representation of the outfit and visual features of both the scenery and person. Their method also provides feedback on how to improve the fashionability score. In [5] neural networks are used to acquire multimodal representations of items based on the item image, category and title, to pool these into one outfit representation and to score the outfit's fashionability. Other approaches to outfit fashionability prediction also exist. In [1] an outfit is treated as an ordered sequence and a bidirectional long short-term memory (LSTM) model is used to learn the compatibility relationships among the fashion items. In [4] the visual compatibility of clothing items is captured with a correlated topic model to automatically create capsule wardrobes. [6] build an end-to-end learning framework that improves item recommendation with co-supervision of item generation. Given an image of a top and a description of the requested bottom (or vice versa) their model composes outfits consisting of one top piece and one bottom piece.

None of the above approaches work with region-level and word-level representations, nor make use of an attention mechanism. In contrast, we infer which product features are most important for the outfit recommendation task through the use of an attention mechanism on regions and words.

3 METHODOLOGY

Section 3.1 describes our baseline model, which fuses the visual and textual information with standard common space fusion. Next, Section 3.2 elaborates our model architecture which fuses the visual and textual information through attention.

In all formulas, matrices are written with capital letters and vectors are bolded. We use letters W and $\boldsymbol{b}$ to refer to respectively the weights and bias in linear and non-linear transformations.

3.1 Baseline

Our baseline model is the method of [11]. The model receives two triplets as input: a triplet of image embeddings $(\boldsymbol{x}_{(u)}, \boldsymbol{x}^+_{(v)}, \boldsymbol{x}^-_{(v)})$ of dimension d_i and a triplet of corresponding sentence embeddings $(\boldsymbol{t}_{(u)}, \boldsymbol{t}^+_{(v)}, \boldsymbol{t}^-_{(v)})$ of dimension d_t. How these image and sentence embeddings are obtained is detailed in Section 4.4. Embeddings $\boldsymbol{x}_{(u)}$ and $\boldsymbol{x}^+_{(v)}$ represent images of respectively type u and type v which are compatible. Compatible means that the images represented by $\boldsymbol{x}_{(u)}$ and $\boldsymbol{x}^+_{(v)}$ appear together in some outfit. Meanwhile $\boldsymbol{x}^-_{(v)}$ represents a randomly sampled image of the same type as $\boldsymbol{x}^+_{(v)}$ that has not been seen in an outfit with $\boldsymbol{x}_{(u)}$ and is therefore considered to be incompatible with $\boldsymbol{x}_{(u)}$.

The triplets are first projected to a common, semantic space $\mathcal{S}$ of dimension d_g. The purpose of the common space is to better capture the notions of image similarity, text similarity and image-text similarity. Therefore, three losses are defined on the common space. A visual-semantic loss $\mathcal{L}_{vse}$ enforces that each image should be closer to its own description than to the descriptions of the other images in the triplet:

$$\mathcal{L}_{vse} = \frac{\mathcal{L}_{vse,\boldsymbol{x}_{(u)}} + \mathcal{L}_{vse,\boldsymbol{x}^+_{(v)}} + \mathcal{L}_{vse,\boldsymbol{x}^-_{(v)}}}{3} \tag{1}$$

$$\mathcal{L}_{vse,\boldsymbol{x}_{(u)}} = \frac{\begin{array}{c}\ell(W_i\boldsymbol{x}_{(u)}, W_s\boldsymbol{t}_{(u)}, W_s\boldsymbol{t}^+_{(v)}) \\ + \ell(W_i\boldsymbol{x}_{(u)}, W_s\boldsymbol{t}_{(u)}, W_s\boldsymbol{t}^-_{(v)})\end{array}}{2} \tag{2}$$

$$\text{with } \ell(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \max(0, f(\boldsymbol{x}, \boldsymbol{z}) - f(\boldsymbol{x}, \boldsymbol{y}) + m) \tag{3}$$

$$\text{and } f(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^T \cdot \boldsymbol{y}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{y}||} \tag{4}$$

with $W_i \in \mathbb{R}^{d_g \times d_i}$ and $W_s \in \mathbb{R}^{d_g \times d_t}$ projections to the common space, ℓ the standard triplet loss, m the margin, and f the cosine similarity. $\mathcal{L}_{vse,\boldsymbol{x}^+_{(v)}}$ and $\mathcal{L}_{vse,\boldsymbol{x}^-_{(v)}}$ are computed analogous to Eq. 2. A visual similarity loss $\mathcal{L}_{vsim}$ enforces that an image of type v should be closer to an image of the same type v than to an image of another type u:

$$\mathcal{L}_{vsim} = \frac{\begin{array}{c}\ell(W_i\boldsymbol{x}^+_{(v)}, W_i\boldsymbol{x}^-_{(v)}, W_i\boldsymbol{x}_{(u)}) \\ +\ \ell(W_i\boldsymbol{x}^-_{(v)}, W_i\boldsymbol{x}^+_{(v)}, W_i\boldsymbol{x}_{(u)})\end{array}}{2} \tag{5}$$

with $W_i \in \mathbb{R}^{d_g \times d_i}$ the image projection to the common space and ℓ the standard triplet loss of Eq. 3. Finally, a textual similarity loss $\mathcal{L}_{tsim}$ is defined analogous to Eq. 5.

Next, a type-specific compatibility space $C_{(u,v)}$ of dimension d_c is inferred for each pair of types u and v. In $C_{(u,v)}$ a compatibility loss $\mathcal{L}_{comp}$ enforces that compatible images are closer together than non-compatible images:

$$\mathcal{L}_{comp} = \ell(W_c^{(u,v)}W_i\boldsymbol{x}_{(u)}, W_c^{(u,v)}W_i\boldsymbol{x}^+_{(v)}, W_c^{(u,v)}W_i\boldsymbol{x}^-_{(v)}) \tag{6}$$

with $W_i \in \mathbb{R}^{d_g \times d_i}$ the image projection to the common space, $W_c^{(u,v)} \in \mathbb{R}^{d_c \times d_g}$ the projection associated with $C_{(u,v)}$, and ℓ the standard triplet loss of Eq. 3.

The final training loss is:

$$\mathcal{L} = \mathcal{L}_{comp} + \lambda_1\mathcal{L}_{vsim} + \lambda_2\mathcal{L}_{tsim} + \lambda_3\mathcal{L}_{vse} \tag{7}$$

with λ_1, λ_2 and λ_3 scalar parameters.

3.2 Attention-based Fusion for Outfit Recommendation

The downside of the baseline model is that the item representations are quite coarse and the interaction between the visual and textual modality is quite limited. Instead, we would like to highlight certain parts of an image or words in a description which correspond to important product features for making fashionable item combinations, and integrate this into a multimodal item representation. Therefore we propose an attention-based fusion model, which we obtain by making a few adjustments to the baseline model.

Firstly, the first input to the attention-based fusion model is a triplet of region-level image features $(\boldsymbol{x}_{\mathbf{1:N}(u)}, \boldsymbol{x}^+_{\mathbf{1:N}(v)}, \boldsymbol{x}^-_{\mathbf{1:N}(v)})$ of dimension d_i, where N denotes the number of regions. Depending on the attention mechanism used, the other input is either a triplet of description-level features $(\boldsymbol{t}_{(u)}, \boldsymbol{t}^+_{(v)}, \boldsymbol{t}^-_{(v)})$ as before or a triplet of word-level features $(\boldsymbol{t}_{\mathbf{1:M}(u)}, \boldsymbol{t}^+_{\mathbf{1:M}(v)}, \boldsymbol{t}^-_{\mathbf{1:M}(v)})$ of dimension d_t, where M denotes the number of words. Details on how these features are obtained can be found in Section 4.4. Since $\mathcal{L}_{vsim}$ and $\mathcal{L}_{vse}$ are formulated at the level of full images, we obtain image-level representations by simply taking the average of the region-level representations, i.e., $\boldsymbol{x}_{(u)} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_{\boldsymbol{i}(u)}$. In the same way we obtain description-level representations from word-level representations for $\mathcal{L}_{tsim}$.

Secondly, we use an attention mechanism to fuse the visual and textual information and obtain a triplet $(\boldsymbol{m}_{(u)}, \boldsymbol{m}^+_{(v)}, \boldsymbol{m}^-_{(v)})$ of multimodal item representations. These multimodal item representations are more fine-grained and allow more complex interactions between the vision and language data. Finally, we project these multimodal item representations to the type-specific compatibility spaces.

How we identify important product features depends on the attention mechanism used. Section 3.2.1 describes visual dot product attention. Section 3.2.2 describes stacked visual attention. Finally, Section 3.2.3 discusses a co-attention mechanism. Furthermore, we also experimented with self-attention [12] on the image regions and words, and some other co-attention and multimodal attention mechanisms [7–9], but these did not improve performance.

3.2.1 Visual Dot Product Attention. Given region-level image features $X \in \mathbb{R}^{N \times d_g}$ and description-level features $\boldsymbol{t} \in \mathbb{R}^{d_g}$, visual dot product attention produces attention weights based on the dot product of the representations of the description and each region:

$$a_i = \tanh(\boldsymbol{x_i}) \cdot \tanh(\boldsymbol{t}) \tag{8}$$

with $\boldsymbol{x_i}$ the i'th row of X. Next, the attention weights are normalized and used to compute the visual context vector:

$$\boldsymbol{c} = \sum_{i=1}^{N}\alpha_i\boldsymbol{x_i}, \text{ with } \alpha_i = \text{softmax}([a_1, a_2, ..., a_N])_i \tag{9}$$

with $\boldsymbol{x_i}$ the i'th row of X. The visual context vector $\boldsymbol{c}$ is concatenated with description $\boldsymbol{t}$, i.e., $[\boldsymbol{c}; \boldsymbol{t}]$ with [] the concatenation operator, to obtain a multimodal item representation of dimension $2d_g$.

3.2.2 Stacked Visual Attention. Given region-level image features $X \in \mathbb{R}^{N \times d_g}$ and description-level features $\boldsymbol{t} \in \mathbb{R}^{d_g}$, stacked visual attention [14] produces a multimodal context vector in multiple attention hops, each extracting more fine-grained visual information. In the r'th attention hop, the attention weights and context vector are calculated as:

$$\boldsymbol{a}^{(r)} = \boldsymbol{w}_p^{(r)}\tanh(W_v^{(r)}X^T \oplus (W_t^{(r)}\boldsymbol{q}^{(r-1)} + \boldsymbol{b}_s^{(r)})) \tag{10}$$

$$\boldsymbol{c}^{(r)} = \boldsymbol{\alpha}^{(r)}X, \text{ with } \boldsymbol{\alpha}^{(r)} = \text{softmax}(\boldsymbol{a}^{(r)}) \tag{11}$$

with $W_v^{(r)}, W_t^{(r)} \in \mathbb{R}^{h \times d_g}$ and $\boldsymbol{w}_p^{(r)} \in \mathbb{R}^{1 \times h}$ learnable weights, $\boldsymbol{b}_s^{(r)} \in \mathbb{R}^h$ the bias vector, $\boldsymbol{q}^{(r-1)}$ the query vector from the previous hop, and $\oplus$ the elementwise sum operator. The query vector is initialized to $\boldsymbol{t}$. At the r'th hop, the query vector is updated as:

$$\boldsymbol{q}^{(r)} = \boldsymbol{q}^{(r-1)} + \boldsymbol{c}^{(r)} \tag{12}$$

This process is repeated R times, with R the number of attention hops. Afterwards, the final query vector $\boldsymbol{q}^{(R)}$ is concatenated with description $\boldsymbol{t}$, i.e., $[\boldsymbol{q}^{(R)}; \boldsymbol{t}]$ with [] the concatenation operator, to obtain a multimodal item representation of dimension $2d_g$.

3.2.3 Co-attention. The co-attention mechanism of [15] attends to both the representations of the image regions $X \in \mathbb{R}^{N \times d_g}$ and the representations of the description words $Y \in \mathbb{R}^{M \times d_g}$ as follows.

First, the description words are attended independent of the image regions. The assumption here is that the most relevant words of the description can be inferred independent of the image content, i.e., words referring to color, shape, style and brand can be considered relevant independent of whether they are displayed in the image or not. Given word-level features Y, the textual attention

weights $\boldsymbol{a}^t$ and textual context vector $\boldsymbol{c}^t$ are obtained as:

$$\boldsymbol{a}^t = \underset{in=d_g, out=1, k=1}{\text{Convolution1D}_{t,2}}(\text{ReLU}(\underset{in=d_g, out=d_g, k=1}{\text{Convolution1D}_{t,1}}(Y))) \quad (13)$$

$$\boldsymbol{c}^t = \boldsymbol{\alpha}^t Y, \text{ with } \boldsymbol{\alpha}^t = \text{softmax}(\boldsymbol{a}^t) \quad (14)$$

where *Convolution1D* refers to the 1D-convolution operation with *in* input channels, *out* output channels and kernel size *k*.

Next, the image regions are attended in R attention hops. In the r'th attention hop, the textual context vector $\boldsymbol{c}^t$ is merged with each of the region-level image features in X using multimodal factorized bilinear pooling (MFB). MFB consists of an *expand stage* where the unimodal representations are projected to a higher dimensional space of dimension $p2d_g$ (with p a hyperparameter) and then merged with elementwise multiplication followed by a *squeeze stage* where the merged feature is transformed back to a lower dimension $2d_g$. For a detailed explanation of MFB the reader is referred to [15]. The MFB operation results in a multimodal feature matrix $M \in \mathbb{R}^{N \times 2d_g}$. Then, the visual attention weights $\boldsymbol{a}^{v,(r)}$ and context vector $\boldsymbol{c}^{v,(r)}$ are calculated based on this merged multimodal feature matrix M:

$$\boldsymbol{a}^{v,(r)} = \underset{in=d_g, out=1, k=1}{\text{Convolution1D}^{(r)}_{v,2}}(\text{ReLU}(\underset{in=2d_g, out=d_g, k=1}{\text{Convolution1D}^{(r)}_{v,1}}(M))) \quad (15)$$

$$\boldsymbol{c}^{v,(r)} = \boldsymbol{\alpha}^{v,(r)} M, \text{ with } \boldsymbol{\alpha}^{v,(r)} = \text{softmax}(\boldsymbol{a}^{v,(r)}) \quad (16)$$

The visual context vectors of all hops are concatenated and transformed to obtain the final visual context vector $\boldsymbol{c}^v$:

$$\boldsymbol{c}^v = W_f[\boldsymbol{c}^{v,(1)}; \boldsymbol{c}^{v,(2)}; ...; \boldsymbol{c}^{v,(R)}] \quad (17)$$

with $W_f \in \mathbb{R}^{R2d_g \times 2d_g}$ and [] the concatenation operator. Finally, the final visual context vector $\boldsymbol{c}^v$ is merged with the textual context vector $\boldsymbol{c}^t$ using MFB to acquire a multimodal item representation of dimension $2d_g$.

4 EXPERIMENTAL SETUP

4.1 Experiments and Evaluation

All models are evaluated on two tasks. In the fashion compatibility (FC) task, a candidate outfit is scored based on how compatible its items are with each other. More precisely, the outfit compatibility score is computed as the average compatibility score across all item pairs in the outfit. Since the compatibility of two items is measured with cosine similarity, the outfit compatibility score will lie in the interval $[-1, 1]$. The performance of the FC task is evaluated using the area under a ROC curve (AUC). In the fill-in-the-blank (FITB) task the goal is to select from a set of four candidate items the item which is the most compatible with the remainder of the outfit. More precisely, the most compatible candidate item is the one which has the highest total compatibility score with the items in the remainder of the outfit. Performance for this task is evaluated with accuracy.

FC questions and FITB questions that consist of images without a description are discarded to keep evaluation fair for all models. Also note that if a pair of items have a type combination that was never seen during training, the model has not learned a type-specific compatibility space for that pair. Such pairs are ignored during evaluation. Hence, we also use the training set to determine which pairs of types do not effect outfit fashionability.

4.2 Datasets

We evaluate all models on three different datasets: Polyvore68K-ND, Polyvore68K-D and Polyvore21K.

4.2.1 Polyvore68K. The Polyvore68K dataset[1] [11] originates from Polyvore. Two different train-test splits are defined for the dataset. Polyvore68K-ND contains 53,306 outfits for training, 10,000 for testing, and 5,000 for validation. It consists of 365,054 items, some of which occur both in the training and test set. However, no outfit appearing in one of the three sets is seen in the other two. The other split, Polyvore68K-D, contains 32,140 outfits, of which 16,995 are used for training, 15,145 for testing and 3,000 for validation. It has 175,485 items in total, where no item seen during training appears in the validation or test set. Both splits have their own FC questions and FITB questions.

Each item in the dataset is represented by a product image and a short description. Items have one of 11 coarse types (see Table 2 in Appendix A).

4.2.2 Polyvore21K. Another dataset collected from Polyvore is the Polyvore21K dataset[2] [1]. It contains items of 380 different item types, however not all are fashion related, e.g., furniture, toys, skincare, food and drinks, etc. We delete all items with types unrelated to clothing, clothing accessories, shoes and bags. The remaining 180 types are all fashion related, but some of them are very-fine grained. We make the item types more coarse to avoid an abundance of type-specific compatibility spaces, i.e. more than 5,000, which is unfeasible. The remaining 37 types can be found in Table 2 in Appendix A. Eventually, this leaves 16,919 outfits for training, 1,305 for validation and 2,701 for testing. There are no overlapping items between the three sets. Each item has an associated image and description.

During evaluation we use the FC questions and FITB questions supplied by [11] for the Polyvore21K dataset, after removal of fashion unrelated items.

4.3 Comparison with Other Works

This work uses a slightly different setup than the work of [11] and therefore our results are not exactly comparable with theirs. Firstly, we do not evaluate our models on the same set of FC and FITB questions. This is because we discard questions consisting of images without a description as explained in Section 4.1. Secondly, the item types used for the Polyvore21K dataset are different. It is unclear from [11] how they obtain and use the item types of the Polyvore21K dataset, as these have only been made public recently. In this work, we used the publicly available item types after cleaning as detailed in Section 4.2.2.

4.4 Training Details

All images are represented with the ResNet18 architecture [2] pretrained on ImageNet. More precisely, as in [11] we take the output of the $7 \times 7 \times 256$ *res4b_relu* layer. For the models operating on image regions this results in 49 regions for every image, each with a dimension d_i of 256. For the models working with full images, we use an additional average pooling layer to obtain one image-level

[1] https://github.com/mvasil/fashion-compatibility
[2] https://github.com/xthan/polyvore-dataset

Figure 2: Examples of fill-in-the-blank questions on the Polyvore68K-ND dataset and answers generated by the baseline model and our attention-based fusion model based on stacked visual attention.

representation, also with a dimension d_i equal to 256. The text descriptions are represented with a bidirectional LSTM of which the forward and backward hidden state at timestep M are concatenated, with M the number of words in the descriptions. For models operating on the level of words instead of full descriptions, we concatenate the forward and backward hidden state of the bidirectional LSTM at each timestep j to obtain the representation for the j'th word. The parameters of the ResNet18 architecture and the bidirectional LSTM are finetuned on our dataset during training. Dimensions d_t, d_g, d_c and h are equal to 512. Hyperparameters are set based on the validation set. For the attention mechanisms, the number of attention hops R is set to 2 and hyperparameter p for MFB is set to 2.

All models are trained for 10 epochs using the ADAM optimizer with a learning rate of 5e-5 and a batch size of 128. In the loss functions, factors λ_1 and λ_2 are 5e-5, λ_3 is set to 5e-3 and margin m is 0.2. All models are trained for 5 runs. We do this to counteract the effect of the negative sampling which is done at random during training. To compute performance, we take the average performance on the FC task and FITB task across these 5 runs. In qualitative results, we use a voting procedure to determine the final answer on FC and FITB questions.

	Polyvore68K-ND		Polyvore68K-D		Polyvore21K	
	FC	FITB	FC	FITB	FC	FITB
Common space fusion						
baseline [11]	85.62	56.55	85.07	56.91	86.28	58.35
Attention-based fusion						
visual dot product attention	89.43	61.55	86.85	60.12	88.59	**63.11**
stacked visual attention	**89.68**	**61.92**	**87.25**	**60.48**	**88.89**	62.52
co-attention	89.58	61.20	86.25	59.00	85.04	58.20

Table 1: Results on the fashion compatibility and fill-in-the-blank tasks for the Polyvore68K dataset versions and the Polyvore21K dataset.

5 RESULTS

Table 1 shows the results of the discussed models on the Polyvore68K dataset versions and the Polyvore21K dataset. We outperform standard common space fusion on all three datasets for both the FC and FITB tasks. On the Polyvore68K dataset versions the best results for both tasks are achieved with the fusion method based on stacked visual attention. For the Polyvore21K dataset the best results for the FC task are obtained with the fusion method based on stacked visual attention and for the FITB task with the fusion method based on visual dot product attention. Generally, we observe that a basic attention mechanism such as visual dot product attention obtains

comparable results with more complex attention mechanisms such as stacked visual attention or co-attention.

When focusing on the separate tasks, our attention-based fusion models seem better at distinguishing randomly generated outfits from human-generated outfits than the standard common space fusion models. Especially on the Polyvore68K-ND dataset this observation is apparent. Furthermore, our attended multimodal item representations enable the generation of more fashionable outfits as can be seen from the results on the FITB task. Figure 2 shows some FITB questions and answers generated by the standard common space fusion model and our fusion model based on stacked visual attention for the Polyvore68K-ND dataset. For each of these FITB questions, the ground truth item needs to be selected because of some small details in other items of the outfit which are picked up by our model but not by the baseline. More precisely, for the first example the light blue handbag matches especially well with the light blue clasp of the pump. In the second example, the striped pattern of the handbag returns in the slippers and the yellow of the flower on the handbag returns in the sunglasses. In the third example, the green belt matches well with the green accents in the handbag and mules. In the last example, the T-shirt of an elephant looks nice in combination with the elephant-shaped earrings.

Hence, both quantitative and qualitative results demonstrate that highlighting certain product features in the item representations for making outfit combinations is meaningful and can be achieved with attention.

6 CONCLUSION

In this work we showed that attention-based fusion integrates visual and textual information in a more meaningful way than standard common space fusion. Attention on region-level image features and word-level text features allows to bring certain product features to the forefront in the multimodal item representations, which benefits the outfit recommendation results. We demonstrated this on three datasets, improving over state-of-the-art results on an outfit compatibility prediction task and an outfit completion task.

As future work and to further improve the results, we would like to investigate neural architectures that still better recognise fine-grained fashion attributes in images, to benefit more from the attention-based fusion. Furthermore, we would like to design novel co-attention mechanisms which still better integrate fine-grained visual and textual attributes.

ACKNOWLEDGMENTS

The first author is supported by a grant of the Research Foundation - Flanders (FWO) no. 1S55420N.

REFERENCES

[1] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. 2017. Learning Fashion Compatibility with Bidirectional LSTMs. In *ACM Multimedia*.
[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778.
[3] Ruining He, Charles Packer, and Julian McAuley. 2016. Learning Compatibility Across Categories for Heterogeneous Item Recommendation. In *International Conference on Data Mining*.
[4] Wei-Lin Hsiao and Kristen Grauman. 2018. Creating Capsule Wardrobes From Fashion Images. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 7161–7170.
[5] Yuncheng Li, Liangliang Cao, Jiang Zhu, and Jiebo Luo. 2017. Mining Fashion Outfit Composition Using an End-to-End Deep Learning Approach on Set Data. *IEEE Transactions on Multimedia* 19 (2017), 1946–1955.
[6] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. Improving Outfit Recommendation with Co-supervision of Fashion Generation. In *The World Wide Web Conference (WWW '19)*. ACM, 1095–1105.
[7] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical Question-image Co-attention for Visual Question Answering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*. Curran Associates Inc., 289–297.
[8] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual Attention Networks for Multimodal Reasoning and Matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
[9] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional Attention Flow for Machine Comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.
[10] Edgar Simo-Serra, Sanja Fidler, Francesc Moreno-Noguer, and Raquel Urtasun. 2015. Neuroaesthetics in fashion: Modeling the perception of fashionability.. In *CVPR*. IEEE Computer Society, 869–877.
[11] Mariya I. Vasileva, Bryan A. Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David A. Forsyth. 2018. Learning Type-Aware Embeddings for Fashion Compatibility. In *ECCV*.
[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5998–6008.
[13] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. 2015. Learning Visual Clothing Style with Heterogeneous Dyadic Co-Occurrences. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15)*. IEEE Computer Society, 4642–4650.
[14] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola. 2016. Stacked Attention Networks for Image Question Answering.. In *CVPR*. IEEE Computer Society, 21–29.
[15] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal Factorized Bilinear Pooling with Co-Attention Learning for Visual Question Answering. *IEEE International Conference on Computer Vision (ICCV)* (2017), 1839–1848.

A DATASET ITEM TYPES

Table 2 gives an overview of the different item types in the Polyvore68K dataset and the types that remain in the Polyvore21K dataset after cleaning.

	Item Types
Polyvore68K	Accessories, All body, Bags, Bottoms, Hats, Jewellery, Outerwear, Scarves, Shoes, Sunglasses, Tops
Polyvore21K	Accessories, Activewear, Baby, Bags and Wallets, Belts, Boys, Cardigans and Vests, Clothing, Costumes, Cover-ups, Dresses, Eyewear, Girls, Gloves, Hats, Hosiery and Socks, Jeans, Jewellery, Jumpsuits, Juniors, Kids, Maternity, Outerwear, Pants, Scarves, Shoes, Shorts, Skirts, Sleepwear, Suits, Sweaters and Hoodies, Swimwear, Ties, Tops, Underwear, Watches, Wedding Dresses

Table 2: Item types kept in the Polyvore68K and Polyvore21K datasets.