Explainable Outfit Recommendation with Joint Outfit Matching and Comment Generation

Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, *Member, IEEE*, and Maarten de Rijke

Abstract—Most previous work on outfit recommendation focuses on designing visual features to enhance recommendations. Existing work neglects user comments of fashion items, which have been proven to be effective in generating explanations along with better recommendation results. We propose a novel neural network framework, *neural outfit recommendation* (NOR), that simultaneously provides outfit recommendations and generates abstractive comments. Neural outfit recommendation (NOR) consists of two parts: outfit matching and comment generation. For outfit matching, we propose a convolutional neural network with a mutual attention mechanism to extract visual features. The visual features are then decoded into a rating score for the matching prediction. For abstractive comment generation, we propose a gated recurrent neural network with a cross-modality attention mechanism to transform visual features into a concise sentence. The two parts are jointly trained based on a multi-task learning framework in an end-to-end back-propagation paradigm. Extensive experiments conducted on an existing dataset and a collected real-world dataset show NOR achieves significant improvements over state-of-the-art baselines for outfit recommendation. Meanwhile, our generated comments achieve impressive ROUGE and BLEU scores in comparison to human-written comments. The generated comments can be regarded as explanations for the recommendation results. We release the dataset and code to facilitate future research.

Index Terms—Outfit recommendation, explainable recommendation

1 Introduction

Outfit recommendation plays an increasingly important role in the online retail market.[1] The purpose of outfit recommendation is to promote people's interest and participation in online shopping by recommending fashionable outfits that they may be interested in. Early studies on outfit recommendation are based on small but expert-annotated datasets [1], [2], which prohibits the development of complex models that need large sets of training material (e.g., deep learning-based models). In recent years, with the proliferation of fashion-oriented online communities, e.g., Polyvore[2] and Chictopia,[3] people can share and comment on outfit compositions, as shown in Fig. 1. In addition to a large number of outfit compositions, such crowdsourced data also contains valuable information (e.g., user comments) for building more accurate and intelligent recommender systems.

We address the task of explainable outfit recommendation. Given a top (i.e., upper garment), we need to recommend a short list of bottoms (e.g., trousers or skirts) from a large collection that best match the top and meanwhile generate a sentence for each recommendation so as to explain why the top and the bottom match, and vice versa. By explaining why an outfit is recommended, a recommender system becomes more transparent and trustful, which helps users make faster and better decisions [3]. The task of explainable outfit recommendation is non-trivial because of two main problems: (1) We need to model the compatibility of fashion factors, e.g., color, material, pattern, shape, etc. [4]. (2) We need to model transformations between visual and textual information, which involves mappings from the visual to the textual space.

To address the problems listed above, we propose a neural multi-task learning framework, called *neural outfit recommendation* (NOR). NOR consists of two core ingredients: outfit matching and comment generation. For outfit matching, we employ a convolutional neural network (CNN) with a mutual attention mechanism to extract visual features of outfits. Specifically, we first utilize CNNs to model tops and bottoms as latent vectors; then we propose a mutual attention mechanism that extracts better visual features of both tops and bottoms by employing the top vectors to match the bottom vectors, and vice versa. The visual features are then decoded into a rating score as the matching prediction. For abstractive comment generation, we propose a gated recurrent neural network (RNN) with a cross-modality attention mechanism to transform visual features into a concise sentence. Specifically, for generating a word, NOR learns a mapping between the visual and textual space, which is achieved with a cross-modality attention mechanism. All neural parameters in the two parts of our framework as well as the word embeddings are learned by a multi-task learning approach in an end-to-end back-propagation training paradigm.

There have been several studies on outfit recommendation [1], [2], [5]. The work most similar to ours is by Song et al. [4], who first employ a dual auto-encoder network to

1. http://www.chinainternetwatch.com/19945/online-retail-2020
2. http://www.polyvore.com/
3. http://www.chictopia.com/

- *Y. Lin, Z. Chen, Z. Ren, and J. Ma are with the School of Computer Science and Technology, Shandong University, Qingdao, Shandong 250100, China. E-mail: yu.jie.lin@outlook.com, {chenzhumin, zhaochun.ren, majun}@sdu.edu.cn.*
- *P. Ren and M. de Rijke are with the Informatics Institute, University of Amsterdam, Amsterdam 1000, GG, The Netherlands. E-mail: {p.ren, derijke}@uva.nl.*

Manuscript received 19 June 2018; revised 11 Feb. 2019; accepted 3 Mar. 2019. Date of publication 19 Mar. 2019; date of current version 7 July 2020. (Corresponding author: Zhumin Chen.)
Recommended for acceptance by J. Caverlee.
Digital Object Identifier no. 10.1109/TKDE.2019.2906190

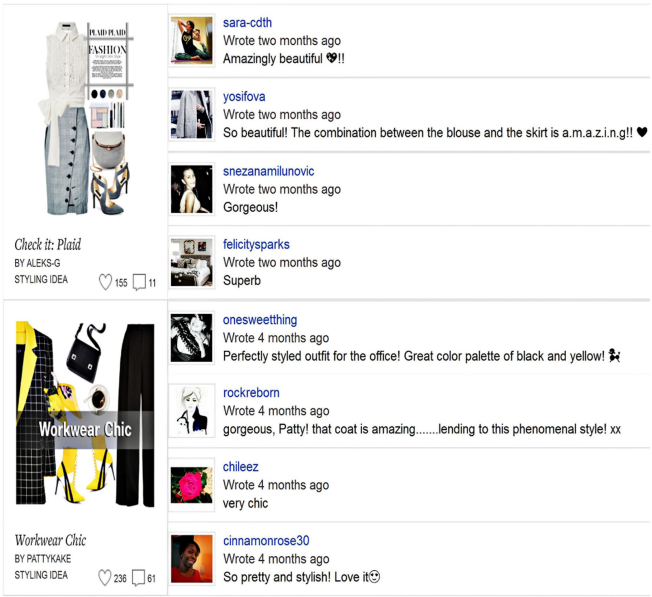


Fig. 1. Outfits with user comments from Polyvore. Users share their outfit compositions with a broad public (left) and others express their comments about the outfit compositions (right).

learn the latent compatibility space, where they jointly model a coherence relation between visual features (i.e., images) and contextual features (i.e., categories, tags). Then they employ Bayesian personalized ranking (BPR) [6] to exploit pairwise preferences between tops and bottoms. The differences between our work and theirs are three-fold. First, our model can not only recommend tops and bottoms, but also generate a readable sentence as a comment. Second, we introduce a mutual and cross-modality attention mechanism to the latent compatibility space instead of a dual auto-encoder network. Third, we jointly train feature extraction and preference ranking in a single back-propagation scheme.

We collect a large real-world dataset from Polyvore.[4] Our dataset contains multi-modal information, e.g., images, contextual metadata of items and user comments, etc. Extensive experimental results conducted on this dataset show that NOR achieves a better performance than state-of-the-art models on outfit recommendation, in terms of AUC, MAP, and MRR. Moreover, comments generated from NOR achieve impressive ROUGE and BLEU scores.

To sum up, our contributions are:

- We explore user comments for improving outfit recommendation quality along with explanations.
- We propose a deep learning based framework named NOR that can simultaneously yield outfit recommendations and generate abstractive comments with good linguistic quality simulating public experience and feelings.
- We use mutual attention to model the compatibility between fashion items and cross-modality attention to model the transformation between the visual and textual space.
- Our proposed approach is shown to be effective in experiments on an existing dataset and a purpose-built large-scale dataset.

2 Related Work

No previous work has studied the task of explainable outfit recommendation by generating natural language comments as explanations. We briefly survey related work on outfit recommendation, on explainable recommendation and on text generation, respectively.

2.1 Outfit Recommendation

Given a photograph of a fashion item (e.g., tops), an outfit recommender system attempts to recommend a photograph of other fashion items (e.g., bottoms). There have been a handful of attempts to solve the task. Iwata et al. [1] propose a probabilistic topic model to recommend tops for bottoms by learning information about coordinates from visual features in each fashion item region. Liu et al. [2] study both outfit and item recommendation problems. They propose a latent Support Vector Machine model for occasion-oriented outfit recommendation, that is, given a user-input occasion, suggesting the most suitable clothing, or recommending items to pair with the reference clothing. Jagadeesh et al. [7] propose two classes of outfit recommenders, namely deterministic and stochastic, while they mainly focus on color modeling for outfit recommendation.

The studies listed above are mostly based on a small, manually annotated dataset, which prevents the development of complex models. Several recent publications have resorted to other sources, where rich data can be harvested automatically, e.g., in the area of personalized whole outfit recommendation. Hu et al. [5] propose a functional tensor factorization method to model interactions between users and fashion items over a dataset collected from Polyvore. McAuley et al. [8] employ a general framework to model human visual preference for a pair of objects from the Amazon co-purchase dataset; they extract visual features with CNNs and introduce a similarity metric to uncover visual relationships. Similarly, He and McAuley [9] introduce a matrix factorization approach that incorporates visual signals into predictors of people's opinions. To take contextual information (such as titles and categories) into consideration, Li et al. [10] classify a given outfit as popular or non-popular through a multi-modal and multi-instance deep learning system. To aggregate multi-modal data of fashion items and contextual information, Song et al. [4] first employ an auto-encoder to exploit their latent compatibility space. Then, they employ Bayesian personalized ranking to exploit pairwise preferences between tops and bottoms. Kang et al. [11] use CNNs to learn image representations and jointly employ collaborating filtering to recommend fashion items for users. Han et al. [12] propose to jointly learn the visual-semantic embeddings and the compatibility relationships among fashion items in an end-to-end manner. They train a bidirectional LSTM model to sequentially predict the next item conditioned on previous ones to learn their compatibility relationships. Song et al. [13] consider fashion domain knowledge for clothing matching and propose a teacher-student scheme to integrate it with neural networks. And they also introduce an attentive scheme to assign rule confidence in the knowledge distillation procedure. Lin et al. [14] introduce an extra supervision signal by forcing the model to generate a bottom image for a given top image, which makes the model extract better visual features.

Even though there is a growing number of studies on outfit recommendation, none of them takes user comments

4. http://www.polyvore.com/

into account and none can give both recommendations and readable comments like we do in this paper.

2.2 Explainable Recommendation

Explainable recommendation not only provides a ranked list of items, but also gives explanations for each recommended item.

Existing work on explainable recommendation can be classified into different categories, depending on the definition of explanation used. Here, we only survey the most closely related studies. Vig et al. [15] propose an explainable recommendation method that uses community tags to generate explanations. Zhang et al. [16] propose an explicit factor model to predict ratings while generating feature-level explanations about why an item is or is not recommended. He et al. [17] propose TriRank and integrate topic models to generate latent factors for users and items for review-aware recommendation. Ribeiro et al. [18] propose LIME, a novel explanation technique that explains the predictions of any classifier in an interpretable and faithful manner, by learning an interpretable model locally around an individual prediction. Ren et al. [19] propose a richer notion of explanation called viewpoint, which is represented as a tuple of a conceptual feature, a topic and a sentiment label; though they provide explanations for recommendations, the explanations are simple tags or extracted words or phrases. Wang et al. [20] develop a multi-task learning solution that uses joint tensor factorization to model user preferences for recommendations and opinionated content for explanations; the algorithm can generate explanations by projecting the features and opinionated phrases onto the space jointly spanned by the user and item factors. In contrast, we generate concise sentences that express why we recommend an outfit based on all user comments. We believe that simulating users to generate comments is a better way to be closer to the user's perspective that fully expresses the users' experience and feelings, making it easier for users to understand and accept.

Some recent work generates text as explanations while providing recommendations. Ni et al. [21] jointly perform personalized recommendation and review generation by combining collaborative filtering with LSTM-based generative models. Li et al. [22]'s work is most similar to ours. By introducing RNNs into collaborative filtering, they jointly predict ratings and generate tips, which express the sentiment of users while reviewing an item. Our work and previous ones differs in four ways. First, we target a different task, i.e., we focus on outfit recommendation not score rating. Second, the recommendation and generation in this paper are not personalized. We determine whether the outfit is matched based on the public perspective. Because the factors that influence people's selections of clothes mainly include the current fashion, occupation, age and regionalism, and we believe that people with similar ages and popularity are usually similar on these factors. Furthermore the generated comments learned from multiple comments (from different online users) reflect a general and common opinion on behalf of multiple users instead of a single specific user. Third, unlike Ni et al. [21]'s work and Li et al. [22]'s work, our task involves multiple modalities (i.e., image and text). Fourth, instead of using a simple RNN, we propose a more complex cross-modality attention mechanism to handle the mapping from the visual to the textual space.

2.3 Text Generation

Text generation involves a wide variety of tasks and studies, such as text summarization [23], [24], machine translation [25], [26], dialogue systems [27], [28], and image captioning [29], [30]. We list some related works on comment or review generation as follows.

Cao et al. [31] present a framework to automatically collect relevant microblogs from microblogging websites to generate comments for popular news on news websites. Lipton et al. [32] design a character-level RNN to generate reviews. The generated reviews are based on auxiliary information, such as user/item IDs, categories and ratings. Radford et al. [33] also train a character-level RNN language model on the Amazon review dataset, which has only one single multiplicative LSTM layer with 4,096 hidden units. They introduce a special unit among the hidden units that can control the sentiment of the generated reviews. Dong et al. [34] propose an attribute-to-sequence model to generate product reviews for given attribute information including user, product and ratings. They first use an attribute encoder to learn representations of the input attributes. Then they employ a stacked LSTM with an attention mechanism to generate reviews based on these representations. Tang et al. [35] propose two novel approaches that first encode contexts, such as sentiment ratings and product ids, into a continuous semantic representation and then decode the semantic representation into reviews with RNNs. Hu et al. [36] combine a variational auto-encoder and a holistic attribute discriminator to generate reviews. They alternately train the auto-encoder and the discriminator. They can dynamically control the attributes of the generated reviews by learning disentangled latent representations with designated semantics. These studies only focus on text generation, and do not jointly perform recommendation.

3 Neural Outfit Recommendation

3.1 Overview

Given a top t_i from a pool $\mathcal{T} = \{t_1, t_2, \ldots, t_{N_t}\}$, the *bottom item recommendation task* is to recommend a ranked list of bottoms from a candidate pool $\mathcal{B} = \{b_1, b_2, \ldots, b_{N_b}\}$. Similarly, the *top item recommendation task* is to recommend a ranked list of tops for a given bottom. The *comment generation task* is to generate a natural-sounding comment c^{tb} for each recommended outfit (i.e., top-bottom pair). The generated comments can be regarded as explanations for each recommended outfit: why is an outfit matched? Note that it does not matter whether we perform bottom item recommendation or top item recommendation, NOR generates similar comments for the same outfit, because the generated comments are for the whole outfit.

As shown in Fig. 2, NOR consists of three core components, a *top and bottom image encoder*, a *matching decoder*, and a *generation decoder*. Based on a convolutional neural network [37], the top and bottom image encoder (Fig. 3a) extracts visual features from images including a pair (t, b), and transforms visual features to the latent representations of t and b, respectively. A mutual attention mechanism is introduced here to guarantee that the top and bottom image encoder can encode the compatibility between t and b into their latent representations. In Fig. 3b, the matching decoder is a multi-layered perceptron (MLP) that evaluates the matching score between t and b. The generation decoder in Fig. 3c is a gated recurrent unit (GRU) [38], which is used to translate the combination of the latent representation of a top

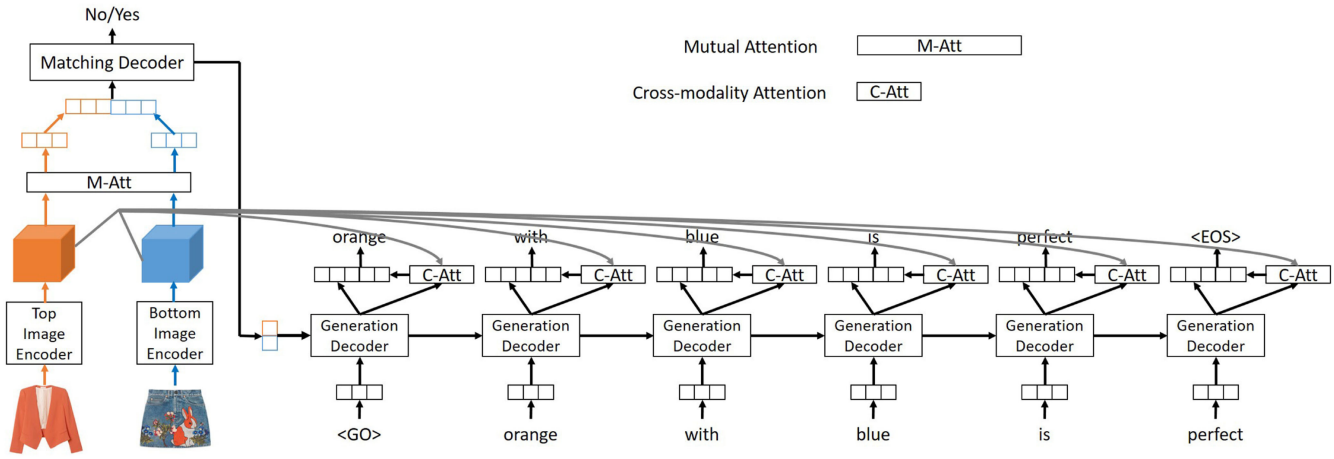
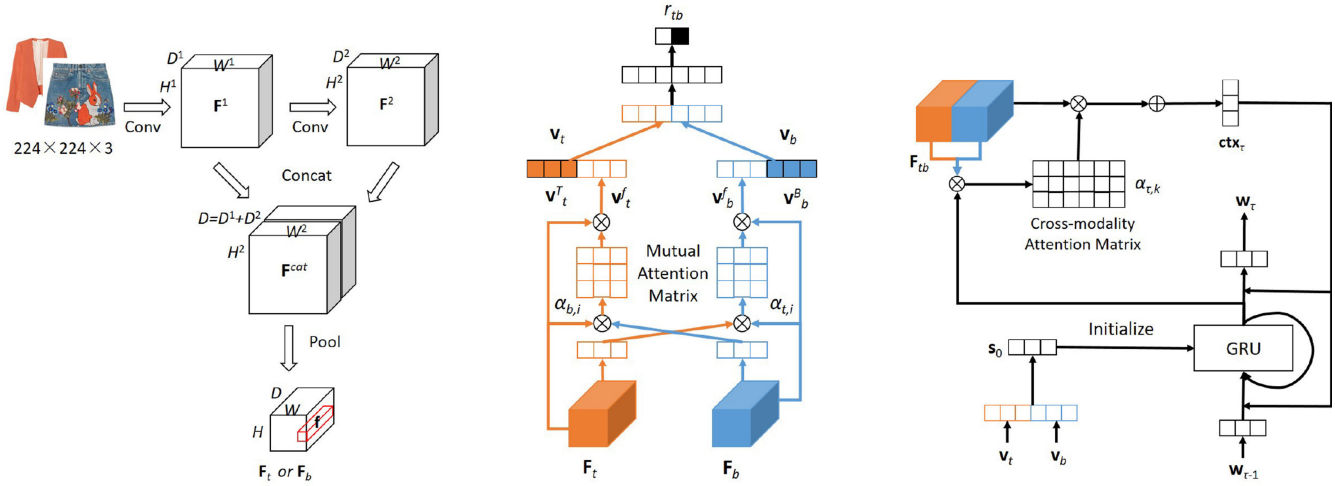


Fig. 2. Overview of the proposed neural outfit recommendation (NOR) architecture. NOR contains three parts: (1) a top and bottom image encoder (corresponding to Fig. 3a), (2) a matching decoder (corresponding to Fig. 3b), and (3) a generation decoder (corresponding to Fig. 3c).



(a) Top and bottom image encoder.

(b) Mutual attention and matching decoder.

(c) Cross-modality attention and generation decoder.

Fig. 3. Details of the neural fashion recommendation architecture (NOR). (a) The top and bottom image encoder extracts visual features $\mathbf{F}_t$ and $\mathbf{F}_b$ from images. (b) Using the mutual attention mechanism, we transform visual features to latent representations $\mathbf{v}_t$ and $\mathbf{v}_b$. Then the matching decoder predicts the matching indicator r_{tb}. (c) At each timestamp τ, the generation decoder employs a cross-modality attention mechanism to generate the word $\mathbf{w}_\tau$.

and the latent representation of a bottom into a sequence of words as comments. For the generation decoder, we propose cross-modality attention to better model the transformation between the visual and textual space.

Next, we detail each of the three core components.

3.2 Top and Bottom Image Encoder

The top image encoder and the bottom image encoder are CNNs, which have been widely used in outfit recommendation [8], [13]. Although there are many powerful architectures, like ResNet [39] or DenseNet [40], training these architectures is not easy, because they have many parameters and need a lot of data and time to train. To balance the training cost and the test performance, we design a two-layer CNNs with mutual attention as the top and bottom image encoder, which has fewer parameters and yields good performance. We verify the effectiveness of our network architecture in experiments.

Given a pair of images (I_t, I_b), we assume that image I_t and image I_b are of size 224×224 with 3 channels. As shown in Fig. 3a, we extract visual features from I_t or I_b via a two-layer CNN. Specifically, we first feed I_t or I_b to a convolutional layer to get primary visual features $\mathbf{F}^1 \in \mathbb{R}^{H^1 \times W^1 \times D^1}$. Then we feed $\mathbf{F}^1$ into another convolutional layer to obtain advanced visual features $\mathbf{F}^2 \in \mathbb{R}^{H^2 \times W^2 \times D^2}$. Given the lessons learned with DensetNet [40] for utilizing visual features more efficiently in different CNN layers, we make sure $H^1 = H^2$ and $W^1 = W^2$ with padding operations so that we can concatenate $\mathbf{F}^1$ and $\mathbf{F}^2$ to get $\mathbf{F}^{cat} \in \mathbb{R}^{H^2 \times W^2 \times (D^1 + D^2)}$. Finally, we use max-pooling in $\mathbf{F}^{cat}$ to obtain the final visual features $\mathbf{F} \in \mathbb{R}^{H \times W \times D}$.

Then we reshape $\mathbf{F} = [\mathbf{f}^1, \ldots, \mathbf{f}^L]$ by flattening the width and height of the original $\mathbf{F}$, where $\mathbf{f}^i \in \mathbb{R}^D$ and $L = W \times H$. We can consider $\mathbf{f}^i$ as the visual features of the ith local region of the input image. Given a pair consisting of a top image I_t and bottom image I_b, they will be fed into the same CNN, i.e., the top and bottom image encoder have the same structure and share parameters. For I_t, the extracted visual features $\mathbf{F}_t$ are denoted as:

$$\mathbf{F}_t = [\mathbf{f}_t^1, \ldots, \mathbf{f}_t^L], \quad \mathbf{f}_t^i \in \mathbb{R}^D. \tag{1}$$

Similarly, for the extracted visual features $\mathbf{F}_b$ of image I_b, we have:

$$\mathbf{F}_b = [\mathbf{f}_b^1, \ldots, \mathbf{f}_b^L], \quad \mathbf{f}_b^i \in \mathbb{R}^D. \tag{2}$$

Previous attention mechanisms [25], [41] are not specifically designed for outfit recommendation, so they are not suitable to model the mutual matching relation between top images and bottom images. We propose mutual attention mechanism to evaluate the correlation and alignment between each local region of I_t and I_b, as shown in Fig. 3b. Mutual attention can model the matching relation from two sides, i.e., from bottom images to top images and from top images to bottom images. So, it is more suitable for outfit recommendation. To calculate the attention weights of top to bottom, we first perform global-average-pooling in $\mathbf{F}_t$, which aggregates the visual features from all local regions to get global visual features $\mathbf{g}_t \in \mathbb{R}^D$ of I_t in

$$\mathbf{g}_t = \frac{1}{L}\sum_{i=1}^{L} \mathbf{f}_t^i. \tag{3}$$

Then, for the ith local region of I_b, we can calculate the attention weight $e_{t,i}$ with $\mathbf{g}_t$ and $\mathbf{f}_b^i$ as in Eq. (4) by following [25]:

$$e_{t,i} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{f}_b^i + \mathbf{U}_a \mathbf{g}_t), \tag{4}$$

where $\mathbf{W}_a$ and $\mathbf{U}_a \in \mathbb{R}^{D\times D}$ and $\mathbf{v}_a \in \mathbb{R}^D$. The attention weights are normalized in:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{i=1}^{L}\exp(e_{t,i})}. \tag{5}$$

Then we calculate the weighted sum of $\mathbf{f}_b^i$ by $\alpha_{t,i}$ to get the attentive global visual features $\mathbf{g}_b^a \in \mathbb{R}^D$ of I_b:

$$\mathbf{g}_b^a = \sum_{i=1}^{L} \alpha_{t,i}\mathbf{f}_b^i. \tag{6}$$

Similarly, we can calculate the attention weights of bottom to top and obtain the attentive global visual features $\mathbf{g}_t^a$ of I_t:

$$\begin{aligned} &\mathbf{g}_b = \frac{1}{L}\sum_{i=1}^{L}\mathbf{f}_b^i, \quad e_{b,i} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{f}_t^i + \mathbf{U}_a \mathbf{g}_b), \\ &\alpha_{b,i} = \frac{\exp(e_{b,i})}{\sum_{i=1}^{L}\exp(e_{b,i})}, \quad \mathbf{g}_t^a = \sum_{i=1}^{L}\alpha_{b,i}\mathbf{f}_t^i. \end{aligned} \tag{7}$$

We then project $\mathbf{g}_t^a$ and $\mathbf{g}_b^a$ to visual feature vectors $\mathbf{v}_t^f$ and $\mathbf{v}_b^f \in \mathbb{R}^{m_v}$:

$$\mathbf{v}_t^f = \mathrm{ReLU}(\mathbf{W}_p \mathbf{g}_t^a), \quad \mathbf{v}_b^f = \mathrm{ReLU}(\mathbf{W}_p \mathbf{g}_b^a), \tag{8}$$

where $\mathbf{W}_p \in \mathbb{R}^{m_v\times D}$ and m_v is the size of $\mathbf{v}_t^f$ and $\mathbf{v}_b^f$.

Finally, building on insights from matrix factorization-based methods [42], [43], [44], we also learn top latent factors $\mathbf{T} \in \mathbb{R}^{N_T\times m_v}$ and bottom latent factors $\mathbf{B} \in \mathbb{R}^{N_b\times m_v}$ directly through which we incorporate collaborative filtering information as a complement to visual features. Specifically, for each top t and each bottom b, we have latent factors $\mathbf{v}_t^T$ and $\mathbf{v}_b^B$:

$$\mathbf{v}_t^T = \mathbf{T}(t,:), \quad \mathbf{v}_b^B = \mathbf{B}(b,:), \tag{9}$$

where $\mathbf{v}_t^T$ and $\mathbf{v}_b^B \in \mathbb{R}^{m_v}$. And we concatenate visual feature vectors and latent factors to get the latent representations $\mathbf{v}_t$ and $\mathbf{v}_b$:

$$\mathbf{v}_t = [\mathbf{v}_t^f, \mathbf{v}_t^T], \quad \mathbf{v}_b = [\mathbf{v}_b^f, \mathbf{v}_b^B], \tag{10}$$

where $\mathbf{v}_t$ and $\mathbf{v}_b \in \mathbb{R}^m$, $m = 2m_v$.

3.3 Matching Decoder

As shown in Fig. 3b, we employ a multi-layer neural network to calculate the matching probability of t and b. Given latent representations $\mathbf{v}_t$ and $\mathbf{v}_b$ calculated in Eq. (10), we first map $\mathbf{v}_t$ and $\mathbf{v}_b$ into a shared space:

$$\mathbf{h}_r = \mathrm{ReLU}(\mathbf{W}_s \mathbf{v}_t + \mathbf{U}_s \mathbf{v}_b), \tag{11}$$

where $\mathbf{h}_r \in \mathbb{R}^n$, and $\mathbf{W}_s$ and $\mathbf{U}_s \in \mathbb{R}^{n\times m}$ are the mapping matrices for $\mathbf{v}_t$ and $\mathbf{v}_b$, respectively. Then we estimate the matching probability as follows:

$$p(r_{tb}) = \mathrm{softmax}(\mathbf{W}_r \mathbf{h}_r), \tag{12}$$

where $\mathbf{W}_r \in \mathbb{R}^{2\times n}$, and $p(r_{tb}) \in \mathbb{R}^2$ which provides the probability distribution in $r_{tb} = 1$ (corresponding to $p(r_{tb} = 1)$) and $r_{tb} = 0$ (corresponding to $p(r_{tb} = 0)$). Here, $r_{tb} = 1$ denotes that t and b match and $r_{tb} = 0$ denotes that t and b do not match. Finally, we can recommend tops or bottoms according to $p(r_{tb})$.

3.4 Generation Decoder

Following existing studies [21], [22], we also use RNNs to generate comments. As shown in Fig. 3c, we employ a GRU with cross-modality attention as the generation decoder. First, we compute the initial hidden state $\mathbf{s}_0$ for the generation decoder with $\mathbf{v}_t$ and $\mathbf{v}_b$ in

$$\mathbf{s}_0 = \tanh(\mathbf{W}_i \mathbf{v}_t + \mathbf{U}_i \mathbf{v}_b), \tag{13}$$

where $\mathbf{s}_0 \in \mathbb{R}^q$, $\mathbf{W}_i$ and $\mathbf{U}_i \in \mathbb{R}^{q\times m}$, and q is the hidden size of the GRU. Then, at each time stamp τ, the GRU reads the previous word embedding $\mathbf{w}_{\tau-1}$, the previous context vector $\mathbf{ctx}_{\tau-1}$ and the previous hidden state $\mathbf{s}_{\tau-1}$ as input to compute the new hidden state $\mathbf{s}_\tau$ and the current output $\mathbf{o}_\tau$ in

$$\mathbf{s}_\tau, \mathbf{o}_\tau = \mathrm{GRU}(\mathbf{w}_{\tau-1}, \mathbf{ctx}_{\tau-1}, \mathbf{s}_{\tau-1}), \tag{14}$$

where $\mathbf{w}_{\tau-1} \in \mathbb{R}^e$, $\mathbf{ctx}_{\tau-1} \in \mathbb{R}^D$, $\mathbf{s}_\tau$ and $\mathbf{o}_\tau \in \mathbb{R}^q$, and e is the word embedding size. The context vector $\mathbf{ctx}_\tau$ for the current timestamp τ is the weighted sum of all visual features from $\mathbf{F}_t$ and $\mathbf{F}_b$ and computed through the cross-modality attention. It matches the current state $\mathbf{s}_\tau$ with each element of $\mathbf{F}_t$ and $\mathbf{F}_b$ to get an importance score which makes better use of the extracted visual features to generate comments by paying attention to particularly effective visual features. Recall that $\mathbf{F}_t = [\mathbf{f}_t^1, \ldots, \mathbf{f}_t^L]$ and $\mathbf{F}_b = [\mathbf{f}_b^1, \ldots, \mathbf{f}_b^L]$; we put them together as follows:

$$\mathbf{F}_{tb} = [\mathbf{f}_{tb}^1, \ldots, \mathbf{f}_{tb}^{2L}], \quad \mathbf{f}_{tb}^i \in \mathbb{R}^D. \tag{15}$$

The context vector $\mathbf{ctx}_\tau$ is then computed by following [41]:

$$\begin{aligned} &e_{\tau,k} = \mathbf{s}_\tau^\top \mathbf{W}_g \mathbf{f}_{tb}^k, \quad \alpha_{\tau,k} = \frac{\exp(e_{\tau,k})}{\sum_{k=1}^{2L}\exp(e_{\tau,k})}, \\ &\mathbf{ctx}_\tau = \sum_{k=1}^{2L}\alpha_{\tau,k}\mathbf{f}_{tb}^k, \end{aligned} \tag{16}$$

where $\mathbf{W}_g \in \mathbb{R}^{q\times D}$. Then $\mathbf{o}_\tau$ and $\mathbf{ctx}_\tau$ are used to predict the τth word in Eq. (17):

$$p(\mathbf{w}_\tau|\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{\tau-1}) = \text{softmax}(\mathbf{W}_o\mathbf{o}_\tau + \mathbf{U}_o\mathbf{ctx}_\tau), \quad (17)$$

where $\mathbf{W}_o \in \mathbb{R}^{|V|\times q}$, $\mathbf{U}_o \in \mathbb{R}^{|V|\times D}$, and V is the vocabulary.

3.5 Multi-Task Learning Framework

We use negative log-likelihood (NIL) for both the matching task and generation task. For the matching task, we define the loss function as follows:

$$L_{mat} = \sum_{\{r_{tb}\,|\,(t,b)\in\mathcal{P}^+\cup\mathcal{P}^-\}} -\log p(r_{tb}), \quad (18)$$

where $\mathcal{P}^+ = \{(t_{i_1}, b_{j_1}), (t_{i_2}, b_{j_2}), \ldots, (t_{i_N}, b_{j_N})\}$, $t_i \in \mathcal{T}$, $b_j \in \mathcal{B}$ is the set of positive combinations, which are top-bottom pairs extracted from the outfit combinations on Polyvore. $\mathcal{P}^- = \{(t,b)\,|\,t \in \mathcal{T}, b \in \mathcal{B} \wedge (t,b) \notin \mathcal{P}^+\}$ is the set of negative combinations, which are formed by tops and bottoms sampled randomly. Here, for positive combinations, $p(r_{tb})$ means the probability of $p(r_{tb} = 1)$, i.e., the given pair matches; for negative pairs, $p(r_{tb})$ means the probability of $p(r_{tb} = 0)$, i.e., the given pair does not match.

As for the generation task, the loss function is defined in

$$L_{gen} = \sum_{\{c_k^{tb}\,|\,c_k^{tb}\in\mathcal{C}^{tb}\wedge(t,b)\in\mathcal{P}^+\}} -\log p(c_k^{tb}), \quad (19)$$

where $\mathcal{C}^{tb} = \{c_1^{tb}, c_2^{tb}, \ldots, c_{N_{tb}}^{tb}\}$ is the set of comments for each positive combinations of top t and bottom b. Note that we ignore the generation loss for negative combinations. We also add L2 loss as regularization to avoid overfitting:

$$L_{reg} = \|\Theta\|_2^2, \quad (20)$$

where Θ is the set of neural parameters. Finally, the multi-task objective function is a linear combination of L_{mat}, L_{gen} and L_{reg}:

$$L = L_{mat} + L_{gen} + \lambda_{reg}L_{reg}, \quad (21)$$

where λ_{reg} is used to adjust the weight of the regularization term. The whole framework can be efficiently trained using back-propagation in an end-to-end paradigm.

4 Experimental Setup

We set up experiments aimed at assessing the recommendation and generation performance; details shared between the two experiments are presented below.

4.1 Datasets

In this section, we briefly introduce existing datasets and detail how we build our own dataset, *ExpFashion*.

Existing fashion datasets include *WoW* [2], *Exact Street2Shop* [45], *Fashion-136K* [7], and *FashionVC* [4] datasets. *WoW*, *Exact Street2Shop*, and *Fashion-136K* are collected from street photos[5] and thus inevitably involve a clothing parsing technique, which still remains a great challenge in the computer vision domain [4], [46], [47]. Even though *FashionVC* is crawled from Polyvore, it lacks user comments. Moreover, the small scale of all existing datasets makes them insufficient for text generation. We employ *FashionVC* only to evaluate the recommendation part.

TABLE 1
Dataset Statistics

Dataset	Tops	Bottoms	Outfits	Comments
WoW [2]	17,890	15,996	24,417	–
Exact Street2Shop [45]	–	–	39,479	–
Fashion-136K [7]	–	–	135,893	–
FashionVC [4]	14,871	13,663	20,726	–
ExpFashion	29,113	20,902	200,745	1,052,821

To be able to evaluate the recommendation and generation results, we collected a large dataset from Polyvore. In particular, starting from 1,000 seed outfits, we crawled new outfits given an item from existing outfits, and stored outfits in the dataset, iteratively. To balance quality and quantity, we only considered outfits with comments longer than 3 words. We also removed tops or bottoms with fewer than 3 occurrences. We ended up with 200,745 outfits with 29,113 tops, 20,902 bottoms, and 1,052,821 comments. We randomly selected 1,000 tops and bottoms as validation set, 2,000 tops and bottoms as test set, and the remainder as training set. Most selected tops and bottoms with their positive bottoms and tops in validation and test set are unpopular, and are not seen in training set. Here, "unpopular" means that the positive combinations of a top or a bottom is less than 10. Since it is time consuming to evaluate each top-bottom pair, we followed existing studies [4] and randomly selected bottoms to generate 100 candidates along with the positive bottoms for each top in validation and test set. For each top in the validation or test set, the positive bottoms are those that have been matched with the top on Polyvore, which form our ground truth for recommendation; they should be more in line with fashion than other candidates. The same is true for both bottom item recommendation and top item recommendation. For the generation task, we use all actual user comments of each outfit in the *ExpFashion* dataset as the references to evaluate the generated comments. The statistics of *ExpFashion* are listed in Table 1; for comparison, we also describe datasets used in previous work.

We also harvested other domains of information, e.g., visual images, categories, title description, etc., and other kinds of items, e.g., shoes, accessories, etc. All this information can be employed for future research.[6] Because there is no user information (ID or any other information) in the datasets, none of the models (the baselines and our own model) in both recommendation and generation are personalized models. There are no labels in the ExpFashion dataset that identify whether a comment is for the top or bottom; most user comments are for the whole top-bottom pairs, so all models generate comments for complete outfits.

4.2 Implementation Details

For the networks in the top and bottom image encoder, we set the kernel size of all convolutional layers to 3×3, the stride to 1, the padding to 1, the activation function to relu, and the pooling size to 16×16. As a result, we have $H^1 = H^2 = W^1 = W^2 = 224$, $D^1 = D^2 = 32$, $H = W = 14$ and $D = D^1 + D^2 = 64$. The latent representation size m is searched in $[200, 400, 600]$, but there is no significant difference; we set m to 600. For the matching decoder, we set the shared space size n to 256. The input and output

5. http://www.tamaraberg.com/street2shop/

6. The dataset is available at https://bitbucket.org/Jay_Ren/fashion_recommendation_tkde2018_code_dataset

vocabularies are collected from user comments, which have 92,295 words. We set the word embedding size e to 300 and all GRU hidden state sizes q to 512. The regularization weight λ_{reg} is searched in $[0.00001, 0.0001, 0.001, 0.01]$, where 0.0001 is the best. During training, we initialize model parameters randomly using the Xavier method [48]. We use Adam [49] as our optimization algorithm. For the hyper-parameters of the Adam optimizer, we set the learning rate α = 0.001, two momentum parameters $\beta 1$ = 0.9 and $\beta 2$ = 0.999 respectively, and $\epsilon = 10^{-8}$. We also apply gradient clipping [50] with range $[-5, 5]$ during training. To both speed up the training and converge quickly, we use mini-batch size 64 by grid search. We test the model performance on the validation set for every epoch. Because there is no negative outfit in the dataset, we randomly sample a top or bottom for each positive outfit. For negative samples, we do not train the comment generation part. During testing, for comment generation, we use beam search [51] to get better results. To avoid favoring shorter outputs, we average the ranking score along the beam path by dividing it by the number of generated words. To balance decoding speed and performance, we set the beam size to 3. Our framework is implemented in Tensorflow [52]; the code is available at https://bitbucket.org/Jay_Ren/fashion_recommendation_tkde2018_code_dataset. All experiments were conducted on a single Titan X GPU.

5 Bottom and Top Item Recommendation

In this section, we present our experimental results on the recommendation task. We first specify the experimental details for this task. Then we discuss experimental results on the ExpFashion and FashionVC datasets, respectively.

5.1 Methods Used for Comparison

We consider the following baselines in the bottom and top item recommendation experiments.

- *POP*: POP simply selects the most popular bottoms for each top and vice versa. Here, "popularity" is defined as the number of tops that have been paired with the bottom. POP is frequently used as a baseline in recommender systems [53].
- *NRT*: NRT [22] introduces recurrent neural networks into collaborative filtering. It can jointly predict ratings and generate tips based on latent factors of users and items. For comparison, we adapt NRT to make it compatible with outfit recommendation. The input of NRT are the IDs of a top and a bottom, and the output are the comments for this top-bottom pair, and the matching score between the given top and bottom rather than the rating. And the number of hidden layers for the regression part is set to 1. The beam size is set to 3. In addition, because there are no reviews in our datasets, we remove the relative part from NRT. Other configurations follow the original paper. We do not only compare the recommendation performance of the models we consider, but also compare the quality of the generated comments, see Section 6.1.
- *DVBPR*: DVBPR [11] employs the CNN-F [54] to learn image representations and jointly recommends fashion items to users by collaborative filtering. We modify DVBPR to make it work on our task. First we use CNN-F to learn the image representations of the given top t and bottom b. Then we calculate the matching score m_{tb} as follows:

$$m_{tb} = \mathbf{v}_t^T \mathbf{v}_b, \tag{22}$$

  where $\mathbf{v}_t$ and $\mathbf{v}_b$ are the image representations learned by CNN-F, whose size are set to 100. Finally we also train DVBPR by BPR loss.
- *SetRNN*: SetRNN [10] trains AlexNet [55] to extract visual features from top images and bottom images. And it adapts an RNN as a pooling model to classify a given outfit as popular or unpopular based on the extracted features. We change its target to predict whether a given outfit is matched or not.
- *IBR*: IBR [8] models the relation between objects based on their visual appearance. This work also learns a visual style space, in which related objects are retrieved using nearest-neighbor search. In experiments, the embedding size of objects is set to 100.
- *BPR-DAE*: BPR-DAE [4] is a content-based neural framework that models the compatibility between fashion items based on the Bayesian personalized ranking framework. BPR-DAE is able to jointly model the coherence relation between modalities of items and their implicit matching preference. We set the latent representation size of items to 512 in experiments.

Note that POP and NRT recommend items based on historical records to count the popularity and learn the latent factors respectively. So they cannot generalize well to new items that lack historical records [56]. But DVBPR, SetRNN, IBR, BPR-DAE and NOR model the matching relation between fashion items based on their image content and learn to recommend by visual features. As a result, they can generalize to new items as long as there are images.

5.2 Evaluation Metrics

We employ three evaluation metrics in the bottom and top item recommendation experiments: *Mean Average Precision* (MAP), *Mean Reciprocal Rank* (MRR), and *Area Under the ROC curve* (AUC). All are widely used evaluation metrics in recommender systems [57], [58], [59].

As an example, in bottom item recommendation, MAP, MRR, and AUC are computed as follows,

$$\text{MAP} = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{1}{rel_i}\sum_{j=1}^{|B|}(P(j) \times rel(j)), \tag{23}$$

where B is the candidate bottom list; $P(j)$ is the precision at cut-off j in the list; rel_i is the number of all positive bottoms for top i; $rel(j)$ is an indicator function equaling 1 if the item at rank j is a positive bottom, 0 otherwise.

$$\text{MRR} = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{1}{rank_i}, \tag{24}$$

where $rank_i$ refers to the rank position of the first positive bottom for the ith top.

$$\text{AUC} = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{1}{|E(i)|}\sum_{(j,k)\in E(i)}\delta(f(t_i, b_j) > f(t_i, b_k)), \tag{25}$$

TABLE 2
Results of Bottom and Top Item Recommendation on the ExpFashion Dataset (%)

Method	Bottom item			Top item		
	MAP	MRR	AUC	MAP	MRR	AUC
POP	5.45	6.45	49.19	6.91	9.16	51.71
NRT	6.36	8.54	49.49	7.74	11.78	50.98
DVBPR	8.55	12.10	57.96	11.08	16.98	60.31
SetRNN	6.60	8.86	51.69	7.07	9.93	51.89
IBR	7.30	9.99	52.60	8.22	12.54	52.39
BPR-DAE	10.09	13.89	61.36	11.51	17.73	61.75
NOR	**11.54**†	**15.38**†	**64.75**†	**13.48**†	**20.83**†	**65.09**†

The superscript † indicates that NOR significantly outperforms BPR-DAE.

TABLE 3
Results of Bottom and Top Item Recommendation on the FashionVC Dataset (%)

Method	Bottom item			Top item		
	MAP	MRR	AUC	MAP	MRR	AUC
POP	4.61	5.50	30.10	3.83	4.62	27.13
DVBPR	7.99	8.82	57.46	7.83	8.66	57.32
SetRNN	6.04	6.45	51.75	5.66	6.27	52.34
IBR	6.29	6.74	53.98	6.68	7.38	52.61
BPR-DAE	8.44	**9.34**	60.62	8.03	8.95	60.05
NOR-CG	**8.50**	9.12	**64.17**†	**9.40**	**10.24**	**65.28**†

The superscript † indicates that NOR-CG significantly outperforms BPR-DAE.

where T is the top collection as queries; $E(i)$ is the set of all positive and negative candidate bottoms for the ith top; $\delta(\alpha)$ is an indicator function that equals 1 if α is true and 0 otherwise.

For significance testing we use a paired t-test with $p < 0.05$.

5.3 Results on the ExpFashion Dataset

The outfit recommendation results of NOR and the competing models on the ExpFashion dataset are given in Table 2. NOR consistently outperforms all baseline methods in terms of MAP, MRR, and AUC metrics on the ExpFashion dataset. From the results in the table, we have five main observations. First, NOR significantly outperforms all baselines; NOR achieves the best result on all metrics. Although IBR and BPR-DAE employ pre-trained CNNs (both use AlexNet [55] trained on ImageNet)[7] to extract visual features from images, they do not fine-tune the CNNs during experiments. However, we use CNNs as a part of our model, namely the top and bottom image encoder, and jointly train them with the matching decoder and generation decoder on the dataset. We believe that this enables us to extract more targeted visual features from images for our task. DVBPR and SetRNN also jointly train CNNs to extract visual features. But for NOR, we incorporate the mutual attention mechanism that explicitly models the compatibility between a top and a bottom; this mechanism allows us to learn more effective latent representations for tops and bottoms; see Section 7.1 for a further analysis. Moreover, NOR can utilize the information of user comments to improve the performance of outfit recommendation. In fact, visual features and user comments are two modalities to explain why a top and a bottom match. NOR captures this information with its multi-task learning model. This multi-task learning setup makes recommendations more accurate; see Section 7.2 for a further analysis.

Second, IBR and BPR-DAE both use a pre-trained CNN to extract visual features as input, but BPR-DAE performs better. IBR only executes a linear transformation, while BPR-DAE uses a more sophisticated compatibility space learned by an autoencoder neural network.

Third, NRT does not perform well on most metrics. One important reason is that our dataset is very sparse, where a top or a bottom only has about 8 positive combinations on average. Under such conditions, NRT, which relies on collaborative filtering, cannot learn effective latent factors [60], [61].

Fourth, the performance of POP is the worst; the reason is that popularity cannot be used to reflect why a top and a bottom are matched. In outfit recommendation, the visual feature plays a more decisive role. Incorporating visual signals directly into the recommendation objective can make recommendation more accurate [11]. Because they all use CNNs to extract visual features, DVBPR, SetRNN, IBR, BRP-DAE, and NOR all outperform POP and NRT.

Fifth, all methods' top item recommendations are better than their bottom item recommendations. This is because in our dataset the average number of positive tops that each bottom has is larger than the average number of positive bottoms that each top has. This makes bottom item recommendation more difficult than top item recommendation.

5.4 Results on the FashionVC Dataset

In order to confirm the effectiveness of our recommendation part, we also compare NOR-CG, which is NOR without the comment generation part, with POP, DVBPR, SetRNN, IBR and BPR-DAE on the FashionVC dataset; see Table 3. Because there are no comments on FashionVC, we leave out NRT.

From Table 3, we can see that NOR-CG achieves the best performance in terms of the MAP and AUC scores on the bottom item recommendation task and also in terms of the MAP, MRR and AUC score on the top item recommendation task. NOR is only slightly inferior to BPR-DAE in terms of MRR on the bottom item recommendation. This means that, even without the generation component, NOR-CG can still achieve a better performance than other methods. Our top and bottom image encoder with mutual attention can extract effective visual features for outfit recommendation.

Note that only the differences in terms of AUC are significant. The reason is that the size of the FashionVC dataset is small. Although NOR-CG achieves a 1.37 and 1.29 percent increase in terms of MAP and MRR respectively, it is hard to pass the paired t-test with a small test size.

6 Comment Generation

In this section, we assess the performance of comment generation.

6.1 Methods Used for Comparison

No existing work on outfit recommendation is able to generate abstractive comments. In order to evaluate the performance of NOR and conduct comparisons against meaningful baselines, we refine existing methods to make them capable of generating comments as follows.

7. http://www.image-net.org/

TABLE 4
Results on the Comment Generation Task (%)

Methods	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4			BLEU
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
LexRank	**9.60**	8.88	8.17	**2.51**	2.23	2.09	**9.12**	8.38	7.73	**4.43**	3.65	3.05	30.55
CTR	7.16	11.43	7.95	2.01	2.91	2.17	6.69	10.57	7.39	2.95	5.22	3.10	27.43
RMR	7.46	**12.26**	8.44	2.02	**3.00**	**2.23**	6.91	**11.27**	7.78	2.95	**5.49**	3.22	28.46
NRT	7.75	8.98	7.71	1.80	2.30	1.83	7.52	8.74	7.48	3.05	3.93	2.78	35.61
NOR	9.40†	10.29†	**9.09**†	2.21	2.27	2.05†	8.85†	9.68†	**8.55**†	3.96†	4.26†	**3.33**†	**37.21**†

The superscript † indicates that our model NOR performs significantly better than NRT as given by the 95 percent confidence interval in the official ROUGE script.

- *LexRank*: LexRank [62] is an extractive summarization method. We first retrieve all comments from the training set as a sentence collection. Thereafter, given a top and a bottom, we merge relevant sentence collections into a single document. Finally, we employ LexRank to extract the most important sentence from the document as the comment for this top-bottom pair.
- *CTR*: CTR [63] has been proposed for scientific article recommendation; it solves a one-class collaborative filtering problem. CTR contains a topic model component and it can generate topics for each top and each bottom. For a given top or bottom, we first select the top-30 words from the topic with the highest probability. Then, the most similar sentence from the same sentence collection that is used for LexRank is extracted. For a given outfit of a top and a bottom, we choose the one with the highest degree of similarity from the two extracted sentences of the top and the bottom as the final comment.
- *RMR*: RMR [64] utilizes a topic modeling technique to model review texts and achieves significant improvements compared with other strong topic modeling based methods. We modified RMR to extract comments in the same way as CTR.
- *NRT*: We use the same settings as described above in Section 5.1.

Note that we give an advantage to LexRank, CTR, and RMR, since there are no comments available for many cases both in the experimental environment and in practice.

6.2 Evaluation Metrics

We use ROUGE [65] as our evaluation metric with standard options[8] for the evaluation of abstractive comment generation. It is a classical evaluation metric in the field of text generation [22] and counts the number of overlapping units between the generated text and the ground truth written by users. The ROUGE-N score is defined as follows:

$$ROUGE\text{-}N_{recall} = \sum_{g_n \in \tilde{c}} \frac{C_{co}(g_n)}{\sum_{g_n \in c} C(g_n)}, \tag{26}$$

where $\tilde{c}$ is the generated comment; c is the ground truth comment; g_n is an n-gram; $C(g_n)$ is the number of n-grams in $\tilde{c}$; $C_{co}(g_n)$ is the number of n-grams co-occurring in $\tilde{c}$ and c. $ROUGE\text{-}N_{precision}$ is computed by replacing c with $\tilde{c}$ in $ROUGE\text{-}N_{recall}$. ROUGE-L calculates the longest common subsequence between the generated comment and the true comment. And Rouge-SU4 counts the skip-bigram plus unigram-based co-occurrence statistics. We use Recall, Precision, and F-measure of ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-SU4 to evaluate the quality of the generated comments. We also use BLEU [66] as another evaluation metric, which is defined as follows:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{27}$$

where w_n is the weight of the nth word; p_n is n-gram precision, which is computed as $ROUGE\text{-}N_{precision}$; BP is the brevity penalty:

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{1-r/c}, & \text{if } c \leq r, \end{cases} \tag{28}$$

where c is the length of the generated text and r is the length of the reference text.

6.3 Results

The evaluation results of our model and competing methods on the comment generation task on the ExpFashion dataset are given in Table 4. We report Recall, Precision, and F-measure (in percentage) of ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-SU4. Additionally, we also report BLEU.

Based on the results reported in Table 4, we have three main observations. First, NOR achieves good performance on the ExpFashion dataset. Especially in terms of BLEU and F-measure of ROUGE-1, ROUGE-L and ROUGE-SU4, NOR gets the best results. NOR is not the best performer on all metrics; for example, LexRank has better performance than NOR in terms of ROUGE precision. Also RMR's ROUGE recall is better than NOR. This is because LexRank prefers short sentences while RMR prefers long sentences. In contrast, NOR gets much better ROUGE F-measure and BLEU, which means NOR can generate more appropriate comments. In other words, NOR achieves more solid overall performance than other models. The reasons are two-fold. On the one hand, NOR has a top and bottom image encoder to encode information of visual features into the latent representations of tops and bottoms. So it makes the latent representations in NOR more effective. On the other hand, we employ a mutual attention mechanism to make sure that the generation decoder can better convert visual features into text to generate comments.

Second, one exception to the strong performance of NOR described above is that NOR performs relatively poorly in terms of ROUGE-2. The possible reasons are: (1) The user comments in our dataset are very short, only about 7 words

8. ROUGE-1.5.5.pl -n 4 -w 1.2 -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0.

TABLE 5
Analysis of Attention Mechanisms on the Bottom and Top Item Recommendation Tasks (%)

Attention	Bottom item			Top item		
	MAP	MRR	AUC	MAP	MRR	AUC
NOR-NO	10.96	14.93	64.72	13.38	20.13	65.40
NOR-MA	**12.55**	**16.98**	**67.13**	**14.65**	**21.58**	**66.98**
NOR-CA	11.72	15.48	64.85	13.53	20.71	65.60
NOR-(MA+CA)	11.54	15.38	64.75	13.48	20.83	65.09

in length on average. Naturally, the model trained using this dataset cannot generate long sentences. (2) The mechanism of a typical beam search algorithm makes the model favor short sentences. (3) The extraction-based approaches favor the extraction of long sentences. So with an increase in N in ROUGE-N, the performance of NOR suffers and the superiority of extraction-based methods is clear.

Third, due to the sparsity of the dataset, NRT performs poorly on most metrics.

7 Analysis and Case Study

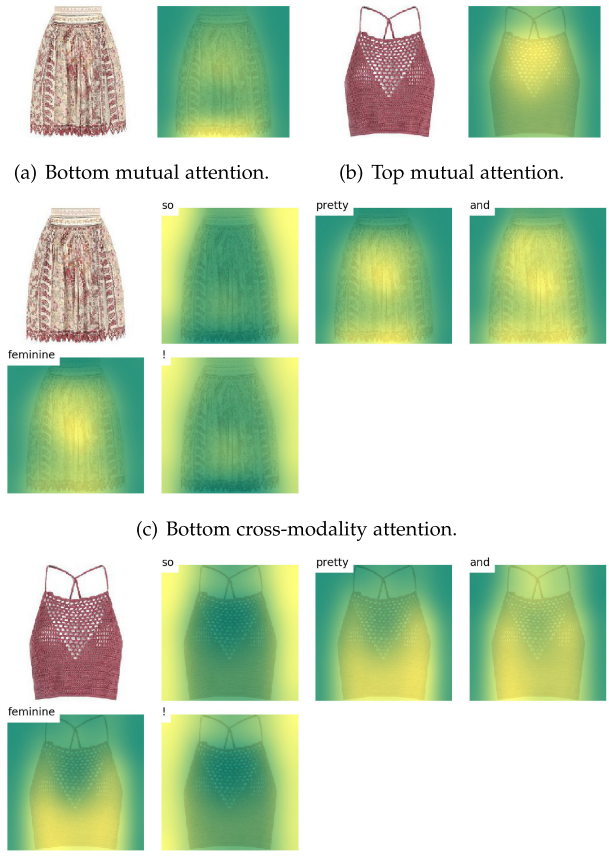
In this section, we conduct further experiments to understand the effectiveness of attention, multi-task learning, and latent factors, followed by recommendation case studies and generation case studies.

7.1 Attention Mechanism Analysis

To verify the effectiveness of the mutual attention mechanism and the cross-modality attention mechanism on the bottom and top item recommendation and comment generation tasks, we conduct experiments with different settings of NOR. The experimental results are shown in Tables 5 and 6.

From Table 5, we notice that NOR-MA (mutual attention only) performs better than NOR-NO (no attention), not only on the bottom item recommendation task, but also on the top item recommendation task. We conclude that the mutual attention mechanism can improve the performance of outfit recommendation. Similarly, as shown in Table 6, we observe that NOR-CA (cross-modality attention only) outperforms NOR-NO. Thus we conclude that the cross-modality attention mechanism is helpful for the comment generation task.

In Table 5, by comparing NOR-MA with NOR-(MA+CA), we also find that NOR-MA outperforms NOR-(MA+CA) on outfit recommendation. That may be because the two kinds of attention mechanism can influence each other through joint training. So in NOR-(MA+CA) the mutual attention mechanism does not reach the same performance as NOR-MA. We think that this performance trade-off is worth making. NOR-(MA+CA) improves over NOR-MA on comment generation in Table 6. Also, NOR-(MA+CA) performs better than NOR-NO on both outfit recommendation and comment generation in Tables 5 and 6, which means that the combination of the two attention mechanisms is effective.



(a) Bottom mutual attention. (b) Top mutual attention. (c) Bottom cross-modality attention. (d) Top cross-modality attention.

Fig. 4. Visualization of mutual attention and cross-modality attention.

We visualize the effects of both attention mechanisms [29], as shown in Fig. 4. For bottom mutual attention, the hem of the skirt gets more attention. And for top mutual attention, NOR pays more attention to the hollow grid on the vest. When generating comments, NOR also knows which words are associated with fashion items. For example, when generating "pretty and feminine," both the top and the bottom get the main attention, because they are the description of the combination. However, for "so" or "!," which are irrelevant to fashion items, NOR pays little attention to the top and the bottom but to the background. So by visualizing attention, we can see that NOR knows how and when to use visual features of tops and bottoms to recommend items and generate comments.

7.2 Multi-Task Learning Analysis

To demonstrate that NOR can use user comments to improve the quality of outfit recommendation by multi-task learning,

TABLE 6
Analysis of Attention Mechanisms on Comment Generation (%)

Attention	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4			BLEU
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
NOR-NO	8.17	9.14	7.99	2.03	2.45	2.00	7.84	8.83	7.69	3.32	4.21	2.99	34.07
NOR-MA	8.00	9.36	8.00	2.10	**2.58**	**2.09**	7.67	9.04	7.69	3.27	4.40	3.03	32.75
NOR-CA	8.42	10.08	8.54	2.12	2.56	**2.09**	7.97	9.59	8.09	3.42	**4.59**	3.18	34.37
NOR-(MA+CA)	**9.40**	**10.29**	**9.09**	**2.21**	2.27	2.05	**8.85**	**9.68**	**8.55**	**3.96**	4.26	**3.33**	**37.21**

TABLE 7
Analysis of Multi-Task Learning (%)

Methods	Bottom item			Top item		
	MAP	MRR	AUC	MAP	MRR	AUC
NOR-CG	10.19	13.65	62.03	12.59	19.17	63.07
NOR	**11.54**	**15.38**	**64.75**	**13.48**	**20.83**	**65.09**

TABLE 8
Analysis of Latent Factors (%)

Methods	Bottom item			Top item		
	MAP	MRR	AUC	MAP	MRR	AUC
NOR-LF	6.55	8.88	49.54	7.83	11.93	50.78
NOR-CG	10.19	13.65	62.03	12.59	19.17	63.07
NOR-WLF	10.62	14.69	62.17	12.28	18.73	62.69
NOR	**11.54**	**15.38**	**64.75**	**13.48**	**20.83**	**65.09**

we compare NOR with NOR-CG; see Table 7. We can see that NOR achieves significant improvements over NOR-CG; on the bottom item recommendation task, MAP increases by 1.35 percent, MRR increases by 1.73 percent, AUC increases by 2.72 percent, and on the top item recommendation task, MAP increases by 0.89 percent, MRR increases by 1.66 percent, AUC increases by 2.02 percent. Through joint learning, our multi-task framework NOR learns shared representations [67] for both recommendation and generation, which can make effective use of the information in comments to improve recommendation performance.

Additionally, by comparing NOR-CG in Table 7 with BPR-DAE in Table 2, we find that, on ExpFashion, NOR-CG also achieves comparable results to other methods, which is consistent with the results on FashionVC (see Section 5.4). On the bottom item recommendation task, NOR-CG achieves 0.10 and 0.67 percent increases in MAP and AUC, respectively; and on the top item recommendation task, NOR-CG achieves a 1.08, 1.44 and 1.32 percent increase in MAP, MRR and AUC, respectively. We conclude that the model structure of NOR in the recommendation part is able to improve recommendation performance.

7.3 Latent Factors Analysis

To analyze the effect of latent factors $\mathbf{T}$ and $\mathbf{B}$ (see Eq. (9)) for recommendation, we compare NOR-LF (NOR that only uses item ID information) with NOR-CG and NOR. As shown in Table 8, we find that NOR-LF does not perform very well. This is because it only uses latent factors to capture the information in the historical matching records and does not take content-based features into consideration. Besides, due to the sparsity of the ExpFashion dataset, matrix factorization-based methods cannot work well. By incorporating the visual information of images, NOR-CG makes up for the deficiency of NOR-LF and improves the recommendation performance. Further, NOR uses the textual information of comments to achieve better recommendations.

(a) Illustration of the bottom item recommendation.

(b) Illustration of the top item recommendation.

Fig. 5. Illustration of the recommendation results. The items highlighted in the red boxes are the positive ones.

TABLE 9
Examples of Recommendations and Generated Comments

			
wow ! this is so beautiful ! love the skirt ! (✓)	love the pink ! (✓)	great denim look . (✓)	love the color combination ! (✓)
			
love this set ! the colours are amazing . (✓)	so beautiful and such a nice style here like it . (✓)	great look great set great color . (✓)	love the red and white ! (✓)
			
great look great set great mixing outfits n ′ nice bag . (✗)	thank you so much for your lovely comments ! (✗)	congrats on top sets sweetie ! xxo . (✗)	great set , love the shoes ! (✗)

Meanwhile, we also compare NOR with NOR-WLF which is NOR without the latent factors. In Table 8, it shows that if there are no the latent factors, the MAP, MRR and AUC of NOR all descend. So we can draw a conclusion that the latent factors can capture the complementary information for the visual features to improve the recommendation performance.

7.4 Recommendation Case Studies

In Fig. 5 we list some recommendation results of NOR on the test set of ExpFashion. For each query item, we select the top-10 recommended items. And we use red boxes to highlight the positive items. Note that even if a recommended item is not highlighted with a red box, it should not be considered negative. We can see that most recommended items are compatible with the query items. For example, the first given top seems to like denim shorts because the positive bottom is a light-colored denim shorts. So the recommended bottoms have many denim shorts or jeans. And the recommended skirts are also reasonable. Because they are short skirts and have similar shape with denim shorts. We also notice that sometimes NOR cannot accurately rank the positive item at the first place. But the recommended items ranked before the positive item are also well enough for the given item, which is reasonable in real applications. For instance, for the last given bottom, the first top looks suitable not only in color but also in texture. Through these examples, we can see that NOR can indeed provide good recommendations.

7.5 Generation Case Studies

For the purpose of analyzing the linguistic quality of generated comments and the correlation between images and comments, we sample some instances from the test set, shown in Table 9. We find that the generated comments are basically grammatical and syntactic. And most of them express feelings and opinions about the combinations from the perspective of the public, which can be treated as explanations about why the top and the bottom match. For example, “wow! this is so beautiful! love the skirt!” shows appreciation to this combination, and “love the skirt” expresses a special preference for the skirt, which is also an appreciation of the outfit. “Love the color combination” points out directly that color matching is the reason of the recommendation. And “so beautiful and such a nice style here i like it” expresses that the style of the outfit is beautiful and nice, which is a good explanation about why recommending this combination. Additionally, NOR generates comments like “great denim look,” where denim is the material of jeans and jackets. Another example is “love the pink,” obviously because the top and the bottom are pink. Similarly, “love the red and white” finds that the top’s color is red and the bottom’s color is white. In summary, NOR is able to generate comments with visual features like texture, color and so on.

There are also some bad cases. For example, “thank you so much for your lovely comments !”, which is feedback on other users’ comments, not a comment posted for the combination. In our datasets, a few comments are communications between users. This indicates that we should study better filtering methods in future work. Other bad cases include statements like “nice bag”. In Polyvore, comments are for outfits, which include not only tops and bottoms, but also shoes, necklaces and so on. So generated comments may include items other than tops and bottoms. These bad cases imply that NOR can generate words not only by visual features but also by ID or other information, which is confirmed when visualizing the effects of attention mechanisms in Section 7.1. There are some other problems we omit here, like duplicate comments or duplicate words, short comments and meaningless comments, which also push us to make further improvements.

8 Conclusions and Future Work

We have studied the task of explainable outfit recommendation. We have identified two main problems: the compatibility of fashion factors and the transformation between visual and textual information. To tackle these problems, we have

proposed a deep learning-based framework, called NOR, which simultaneously gives outfit recommendations and generates abstractive comments as explanations. We have released a large real-world dataset, ExpFashion, including images, contextual metadata of items, and user comments.

In our experiments, we have demonstrated the effectiveness of NOR and have found significant improvements over state-of-the-art baselines in terms of MAP, MRR and AUC. Moreover, we have found that NOR achieves impressive ROUGE and BLEU scores with respect to human-written comments. We have also shown that the mutual attention and cross-modality attention mechanisms are useful for outfit recommendation and comment generation.

Limitations of our work include the fact that NOR rarely generates negative comments to explain why an outfit does not match, that is because most of the comments in the dataset are positive. Furthermore, as short comments take up a large percentage of the dataset, NOR tends to generate short comments.

As to future work, we plan to explore more fashion items in our dataset, e.g., hats, glasses and shoes, etc. Also, to alleviate the problem of generating meaningless comments, studies into coherence in information retrieval [68], [69] or dialogue systems can be explored [70], [71]. We also want to incorporate other mechanisms, such as an auto-encoder, to further improve the performance. Finally, we would like to build a *personalized* outfit recommendation system.

Acknowledgments

The authors thank the anonymous reviewers for their helpful comments. This work is supported by the Natural Science Foundation of China (61672324, 61672322), the Natural Science Foundation of Shandong Province (2016ZRE27468), the Fundamental Research Funds of Shandong University, Ahold Delhaize, the Association of Universities in the Netherlands, and the Innovation Center for Artificial Intelligence (ICAI). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors. Yujie Lin and Pengjie Ren contributed equally.

References

[1] T. Iwata, S. Watanabe, and H. Sawada, "Fashion coordinates recommender system using photographs from fashion magazines," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 2262–2267.

[2] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan, "Hi, magic closet, tell me what to wear!" in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 619–628.

[3] N. Tintarev and J. Masthoff, "A survey of explanations in recommender systems," in *Proc. IEEE 23rd Int. Conf. Data Eng. Workshop*, 2007, pp. 801–810.

[4] X. Song, F. Feng, J. Liu, Z. Li, L. Nie, and J. Ma, "Neurostylist: Neural compatibility modeling for clothing matching," in *Proc. Int. Conf. ACM Multimedia*, 2017, pp. 753–761.

[5] Y. Hu, X. Yi, and L. S. Davis, "Collaborative fashion recommendation: A functional tensor factorization approach," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 129–138.

[6] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.

[7] V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan, "Large scale visual recommendations from street fashion images," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2014, pp. 1925–1934.

[8] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. Int. Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 43–52.

[9] R. He and J. McAuley, "VBPR: Visual bayesian personalized ranking from implicit feedback," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 144–150.

[10] Y. Li, L. Cao, J. Zhu, and J. Luo, "Mining fashion outfit composition using an end-to-end deep learning approach on set data," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1946–1955, Aug. 2017.

[11] W. Cheng Kang, C. Fang, Z. Wang, and J. McAuley, "Visually-aware fashion recommendation and design with generative image models," in *Proc. Int. Conf. Data Mining*, 2017, pp. 207–216.

[12] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional LSTMs," in *Proc. ACM Int. Conf. Multimedia*, 2017, pp. 1078–1086.

[13] X. Song, F. Feng, X. Han, X. Yang, W. Liu, and L. Nie, "Neural compatibility modeling with attentive knowledge distillation," in *Proc. Int. Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 5–14.

[14] Y. Lin, P. Ren, Z. Chen, Z. Ren, J. Ma, and M. de Rijke, "Improving outfit recommendation with co-supervision of fashion generation," in *Proc. Web Conf.*, 2019.

[15] J. Vig, S. Sen, and J. Riedl, "Tagsplanations: Explaining recommendations using tags," in *Proc. Int. Conf. Intell. User Interfaces*, 2009, pp. 47–56.

[16] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proc. Int. Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 83–92.

[17] X. He, T. Chen, M.-Y. Kan, and X. Chen, "Trirank: Review-aware explainable recommendation by modeling aspects," in *Proc. ACM Int. Conf. Inf. Knowl. Manag.*, 2015, pp. 1661–1670.

[18] M. Tulio Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?": Explaining the predictions of any classifier," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[19] Z. Ren, S. Liang, P. Li, S. Wang, and M. de Rijke, "Social collaborative viewpoint regression with explainable recommendations," in *Proc. Int. Conf. Web Search Data Mining*, 2017, pp. 485–494.

[20] N. Wang, H. Wang, Y. Jia, and Y. Yin, "Explainable recommendation via multi-task learning in opinionated text data," in *Proc. Int. Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 165–174.

[21] J. Ni, Z. Lipton, S. Vikram, and J. McAuley, "Estimating reactions and recommending products with generative models of reviews," in *Proc. Int. Joint Conf. Natural Language Process.*, 2017, pp. 783–791.

[22] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam, "Neural rating regression with abstractive tips generation for recommendation," in *Proc. Int. Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 345–354.

[23] P. Li, W. Lam, L. Bing, and Z. Wang, "Deep recurrent generative decoder for abstractive text summarization," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2017, pp. 2091–2100.

[24] Q. Zhou, N. Yang, F. Wei, and M. Zhou, "Selective encoding for abstractive sentence summarization," in *Proc. Annu. Meet. Assoc. Comput. Linguistics*, 2017, pp. 1095–1104.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[27] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 3776–3783.

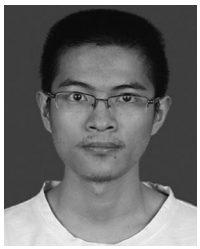
[28] J. D. Williams, K. Asadi, and G. Zweig, "Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning," in *Proc. Annu. Meet. Assoc. Comput. Linguistics*, 2017, pp. 665–677.

[29] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

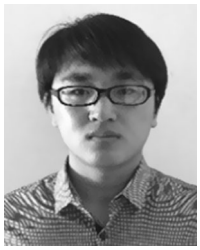
[30] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T. Seng Chua, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6298–6306.

[31] X. Cao, K. Chen, R. Long, G. Zheng, and Y. Yu, "News comments generation via mining microblogs," in *Proc. Int. World Wide Web Conf.*, 2012, pp. 471–472.

[32] Z. C. Lipton, S. Vikram, and J. McAuley, "Capturing meaning in product reviews with character-level generative text models," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1511.03683
[33] A. Radford, R. Jozefowicz, and I. Sutskever, "Learning to generate reviews and discovering sentiment," *CoRR*, 2017. [Online]. Available: https://arxiv.org/abs/1704.01444
[34] L. Dong, S. Huang, F. Wei, M. Lapata, M. Zhou, and K. Xu, "Learning to generate product reviews from attributes," in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 623–632.
[35] J. Tang, Y. Yang, S. Carton, M. Zhang, and Q. Mei, "Context-aware natural language generation with recurrent neural networks," *CoRR*, vol. abs/1611.09900, 2016., http://arxiv.org/abs/1611.09900
[36] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Controllable text generation," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1587–1596.
[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[38] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proc. 8th Workshop Syntax Semantics Structure Statistical Translation*, 2014, pp. 103–111.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[40] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.
[41] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2015, pp. 1412–1421.
[42] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," in *Proc. IEEE Comput. Soc. Press*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[43] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2000, pp. 535–541.
[44] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.
[45] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3343–3351.
[46] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, "Parsing clothing in fashion photographs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3570–3577.
[47] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, "Retrieving similar styles to parse clothing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 1028–1040, May 2015.
[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, 2010.
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[50] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. III–1310–III–1318.
[51] P. Koehn, "Pharaoh: A beam search decoder for phrase-based statistical machine translation models," in *Proc. Conf. Assoc. Mach. Trans. Americas*, 2004, pp. 115–124.
[52] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016., http://arxiv.org/abs/1603.04467
[53] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. Int. World Wide Web Conf.*, 2017, pp. 173–182.
[54] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vision Conf.*, 2014.
[55] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[56] M. Aharon, O. Anava, N. Avigdor-Elgrabli, D. Drachsler-Cohen, S. Golan, and O. Somekh, "Excuseme: Asking users to help in item cold-start recommendations," in *Proc. ACM Conf. Recommender Syst.*, 2015, pp. 83–90.
[57] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proc. ACM Int. Conf. In. Knowl. Manag.*, 2017, pp. 1419–1428.
[58] L. Mei, P. Ren, Z. Chen, L. Nie, J. Ma, and J.-Y. Nie, "An attentive interaction network for context-aware recommendations," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manag.*, 2018, pp. 157–166.
[59] P. Ren, Z. Chen, J. Li, Z. Ren, J. Ma, and M. de Rijke, "Repeatnet: A repeat aware neural recommendation machine for session-based recommendation," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019.
[60] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1763–1777, Jul. 2014.
[61] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 627–636.
[62] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *J. Artif. Intell. Res.*, vol. 22, no. 1, 2004, pp. 457–479.
[63] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2011, pp. 448–456.
[64] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," in *Proc. 8th ACM Conf. Recommender Syst.*, 2014, pp. 105–112.
[65] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Workshop on Text Summarization Branches Out Assoc. Comput. Linguistics*, 2004.
[66] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.
[67] R. Caruana, "Multitask learning," in *Learning to Learn*. Berlin, Germany: Springer, 1998, pp. 95–133.
[68] J. He, W. Weerkamp, M. Larson, and M. de Rijke, "An effective coherence measure to determine topical consistency in user generated content," *Int. J. Document Anal. Recognit.*, vol. 12, no. 3, pp. 185–203, Sep. 2009.
[69] E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. de Rijke, "Mapping queries to the linking open data cloud: A case study using DBpedia," *J. Web Semantics*, vol. 9, no. 4, pp. 418–433, 2011.
[70] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Language Technol.*, 2016, pp. 110–119.
[71] S. Vakulenko, M. de Rijke, M. Cochez, V. Savenkov, and A. Polleres, "Measuring semantic coherence of a conversation," in *Proc. 17th Int. Semantic Web Conf.*, Oct. 2018, pp. 634–651.



Yujie Lin received the BS degree from Shandong University, in 2016. Currently, he is a master's degree student at Shandong University, supervised by Jun Ma. His research area is in information retrieval, recommender systems, and deep learning.



Pengjie Ren is a postdoc researcher with the Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands. His research interests fall in information retrieval, natural language processing, and recommender systems. He has previously published at *TOIS*, SIGIR, AAAI, CIKM, and COLING.

Zhumin Chen received the PhD degree from Shandong University. He is an associate professor with the School of Computer Science and Technology of Shandong University. He is a member of the Chinese Information Technology Committee, Social Media Processing Committee, China Computer Federation Technical Committee (CCF) and ACM. His research interests mainly include information retrieval, big data mining, and processing, as well as social media processing.

Zhaochun Ren received the MSc degree from Shandong University, in 2012, and the PhD degree from the University of Amsterdam, in 2016. He is a professor at Shandong University. He previously was a research scientist at JD.com. Before that he worked as a research associate in University College London. He also worked as a short-term visiting scholar in the Max-Planck-Institut für Informatik in 2012. He is interested in information retrieval, natural language processing, social media mining, and content analysis in e-discovery. He has previously published at SIGIR, ACL, WSDM, CIKM, and KDD.

Jun Ma received the BSc degree from Shandong University, in China, the MSc degree from Ibaraki University, and the PhD degree from Kyushu University, in Japan. He is a full professor at Shandong University. He was a senior researcher with the Department of Computer Science, Ibaraki University, in 1994 and German GMD and Fraunhofer from 1999 to 2003. His research interests include information retrieval, Web data mining, recommender systems, and machine learning. He has published more than 150 papers in international journals and conference papers, including SIGIR, MM, *TOIS*, and *TKDE*. He is a member of the ACM and IEEE.

Maarten de Rijke received the MSc degree in philosophy and mathematics, and the PhD degree in theoretical computer science. He is a university professor in artificial intelligence and information retrieval with the University of Amsterdam. He previously worked as a postdoc at CWI, before becoming a Warwick research fellow at the University of Warwick, United Kingdom. He is the editor-in-chief of the *ACM Transactions on Information Systems*, Springer's *Information Retrieval* book series, and *Foundations and Trends in Information Retrieval*.