Explainable Fashion Recommendation with Joint Outfit Matching and Comment Generation

Lin, Y.; Ren, P.; Chen, Z.; Ren, Z.; Ma, J.; de Rijke, M.

Publication date
2018

Document Version
Final published version

Link to publication

Citation for published version (APA):
Lin, Y., Ren, P., Chen, Z., Ren, Z., Ma, J., & de Rijke, M. (2018). *Explainable Fashion Recommendation with Joint Outfit Matching and Comment Generation*. (v1 ed.) ArXiv. https://arxiv.org/abs/1806.08977v1

Explainable Fashion Recommendation with Joint Outfit Matching and Comment Generation

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

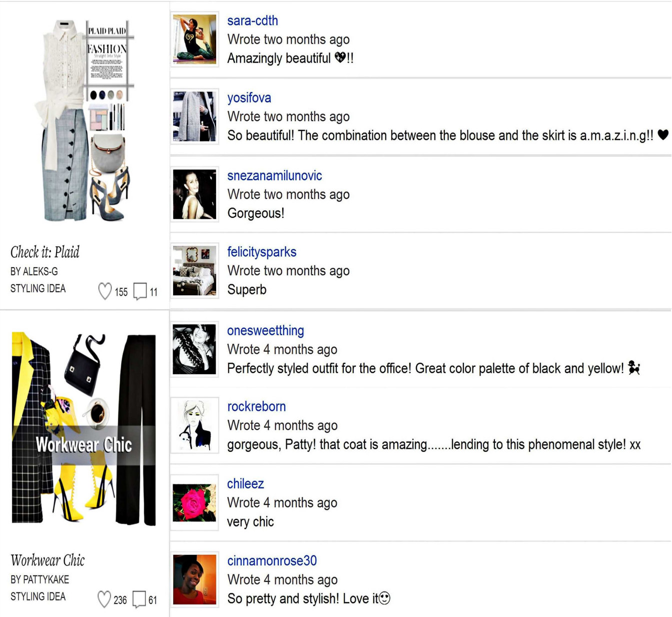
Abstract—Most previous work on fashion recommendation focuses on designing visual features to enhance recommendations. Existing work neglects user comments of fashion items, which have been proved effective in generating explanations along with better recommendation results. We propose a novel neural network framework, *neural fashion recommendation* (NFR), that simultaneously provides fashion recommendations and generates abstractive comments. NFR 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 of outfits. 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 NFR achieves significant improvements over state-of-the-art baselines for fashion 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—Fashion recommendation, explainable recommendation

1 Introduction

Fashion recommendation plays an increasingly important role in the online retail market.[1] The purpose of fashion 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 fashion 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 compatibil-



Check it: Plaid
BY ALEKS-G
STYLING IDEA ♡ 155 □ 11

sara-cdth
Wrote two months ago
Amazingly beautiful 💖!!

yosifova
Wrote two months ago
So beautiful! The combination between the blouse and the skirt is a.m.a.z.i.n.g!! ♥

snezanamilunovic
Wrote two months ago
Gorgeous!

felicitysparks
Wrote two months ago
Superb

Workwear Chic
BY PATTYKAKE
STYLING IDEA ♡ 236 □ 61

onesweetthing
Wrote 4 months ago
Perfectly styled outfit for the office! Great color palette of black and yellow! 🐩

rockreborn
Wrote 4 months ago
gorgeous, Patty! that coat is amazing.......lending to this phenomenal style! xx

chileez
Wrote 4 months ago
very chic

cinnamonrose30
Wrote 4 months ago
So pretty and stylish! Love it😊

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).

- *Yujie Lin, School of Computer Science and Technology, Shandong University, Jinan, China, E-mail: yu.jie.lin@outlook.com*
- *Pengjie Ren, School of Computer Science and Technology, Shandong University, Jinan, China, E-mail: jay.ren@outlook.com*
- *Zhumin Chen, School of Computer Science and Technology, Shandong University, Jinan, China, E-mail: chenzhumin@sdu.edu.cn*
- *Zhaochun Ren, Data Science Lab, JD.com, Beijing, China, E-mail: renzhaochun@jd.com*
- *Jun Ma, School of Computer Science and Technology, Shandong University, Jinan, China, E-mail: majun@sdu.edu.cn*
- *Maarten de Rijke, Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands, E-mail: derijke@uva.nl*
- **Contributed Equally.*

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

ity 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 problems listed above, we propose a neural multi-task learning framework, called the *neural fashion recommendation* (NFR). NFR 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 with a cross-modality attention mechanism to transform visual features into a concise sentence. Specifically, for generating a word, NFR 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 fashion 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 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 advanced 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 NFR achieves a better performance than state-of-the-art models on fashion recommendation, in terms of AUC, MAP, and MRR. Moreover, comments generated from NFR achieve impressive ROUGE and BLEU scores.

To sum up, our contributions can be summarized as follows:

- We explore user comments for improving fashion recommendation quality along with explanations.
- We propose a deep learning based framework named NFR that can simultaneously yield fashion recommendations and generate abstractive comments with good linguistic quality simulating user 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 fashion recommendation. We briefly survey related work on fashion recommendation and on explainable recommendation, respectively.

2.1 Fashion recommendation

Given a photograph of a fashion item (e.g., tops), a fashion 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 fashion 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 fashion recommenders, namely deterministic and stochastic, while they mainly focus on color modeling for fashion recommendation.

The studies listed above are mostly based on a small, manually annotated dataset, which prevents the development of complex models, such as deep learning-based 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 of 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 scalable 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. Han et al. [11] propose to jointly learn a visual-semantic embedding 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.

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

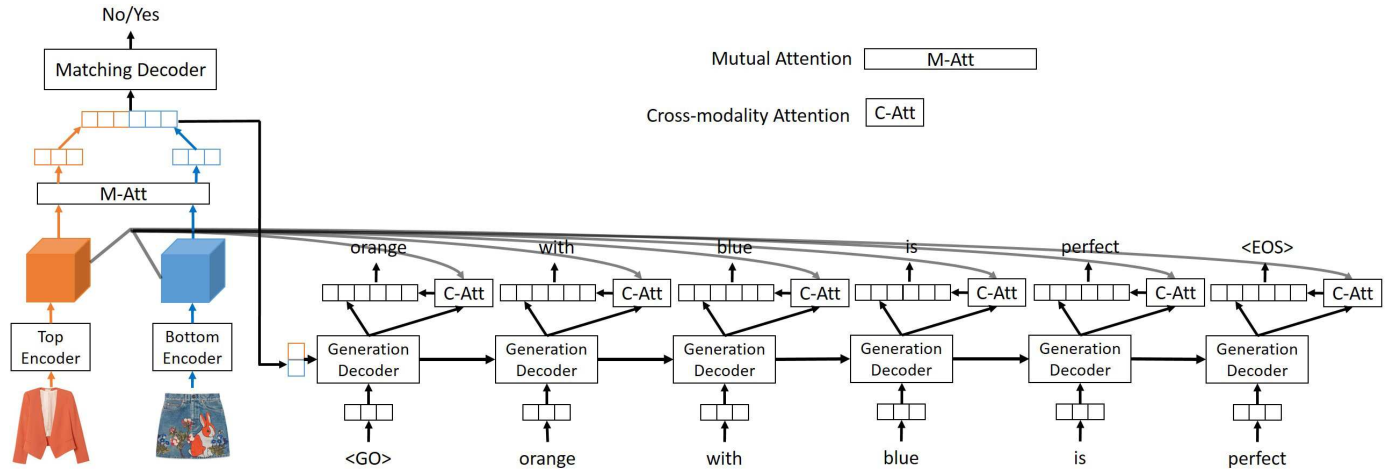


Fig. 2: Overview of the proposed neural fashion recommendation (NFR) architecture. NFR contains three parts: (1) a top and bottom encoder (corresponding to Fig. 3(a)), (2) a matching decoder (corresponding to Fig. 3(b)), and (3) a generation decoder (corresponding to Fig. 3(c)).

Even though there is a growing number of studies on fashion recommendation, none of them takes user comments 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. [12] propose an explainable recommendation method that uses community tags to generate explanations. Zhang et al. [13] 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. [14] propose TriRank and integrate topic models to generate latent factors for users and items for review-aware recommendation. Ribeiro et al. [15] 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. [16] 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. In contrast, we generate concise sentences that express why we recommend an outfit based on all user comments.

Li et al. [17]'s work is most similar to ours. By introducing recurrent neural networks (RNNs) into collaborative filtering, they jointly predict ratings and generate tips, which express the sentiment of users while reviewing an item. Our work differs in three ways. First, we target a different task: they focus on rating prediction while we focus on fashion recommendation. Second, unlike theirs, our task involves multiple modalities (i.e., image and text). Third, instead of using a simple RNN, we propose a more complicated cross-modality attention mechanism to handle the mapping from the visual to the textual space.

3 Neural Fashion Recommendation

3.1 Overview

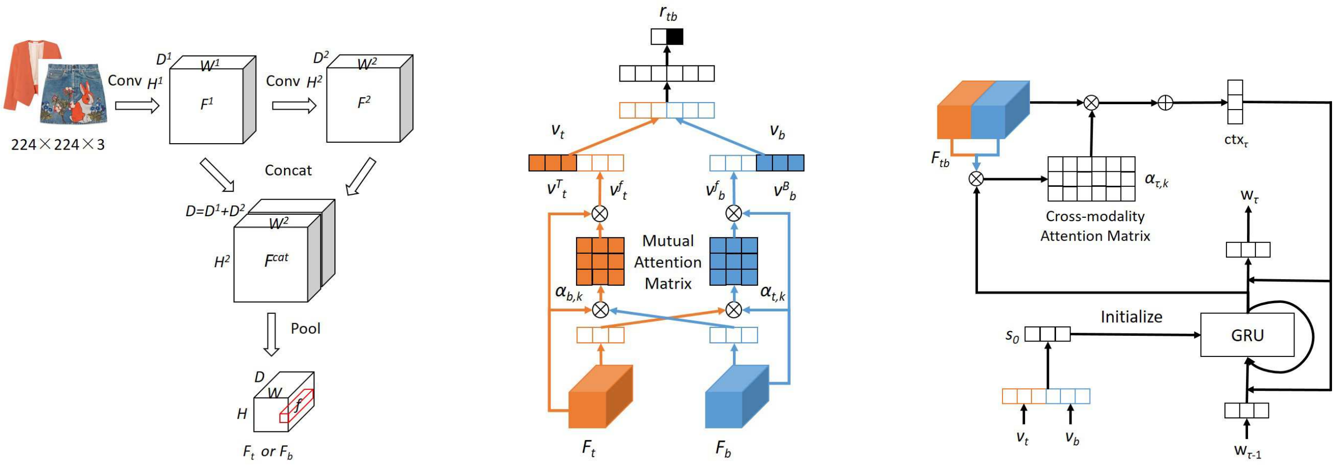
Given a top t_i from a pool $\mathcal{T} = \{t_1, t_2, \ldots, t_{N_t}\}$, the *bottom recommendation task* is to recommend a ranked list of bottoms from a candidate pool $\mathcal{B} = \{b_1, b_2, \ldots, b_{N_b}\}$. Similarly, the *top recommendation task* is to recommend a ranked list of tops for 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 the recommendation results.

As shown in Fig. 2, NFR consists of three core components, a *top and bottom encoder*, a *matching decoder*, and a *generation decoder*. Based on a convolutional neural network [18], the top and bottom encoder (Fig. 3(a)) 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 encoder can encode the compatibility between t and b into their latent representations. In Fig. 3(b), the matching decoder is a multi-layered perceptron (MLP) that evaluates the matching score between t and b. The generation decoder in Fig. 3(c) is a gated recurrent unit (GRU) [19], which is used to translate the combination of the latent representation of a top 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 encoder

The top encoder and the bottom encoder are CNNs. Although there are many powerful architectures, like ResNet [20] or DenseNet [21], we find that two-layer CNNs can achieve good enough performance in our experiments.



(a) Top and Bottom Encoder

(b) Mutual Attention and Matching Decoder

(c) Cross-modality Attention and Generation Decoder

Fig. 3: Details of neural fashion recommendation architecture (NFR). (a) The top and bottom encoder extracts visual features F_t and F_b from images. (b) Using the mutual attention mechanism, we transform visual features to the latent representations v_t and 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 w_τ.

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. 3(a), 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 $F^1 \in \mathbb{R}^{H^1 \times W^1 \times D^1}$. Then we feed F^1 into another convolutional layer to get advanced visual features $F^2 \in \mathbb{R}^{H^2 \times W^2 \times D^2}$. We make sure $H^1 = H^2$ and $W^1 = W^2$ with padding operations so that we can concatenate F^1 and F^2 to get $F^{cat} \in \mathbb{R}^{H^2 \times W^2 \times (D^1+D^2)}$. Finally, we use max-pooling in F^{cat} to obtain the final visual features $F \in \mathbb{R}^{H \times W \times D}$.

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

$$F_t = [f_t^1, \dots, f_t^L], \quad f_t^i \in \mathbb{R}^D. \tag{1}$$

Similarly, for the extracted visual features F_b of image I_b, we have:

$$F_b = [f_b^1, \dots, f_b^L], \quad f_b^i \in \mathbb{R}^D. \tag{2}$$

To better model the compatibility between top image I_t and bottom image I_b, we propose a mutual attention mechanism by evaluating the correlation and alignment between each local region of I_t and I_b, as shown in Fig. 3(b). To calculate the attention weights of top to bottom, we first perform global-average-pooling in F_t to get global visual features g_t of I_t in Eq. 3:

$$g_t = \frac{1}{L} \sum_{i=1}^{L} f_t^i. \tag{3}$$

Then, for the i-th local region of I_b, we can calculate the attention weight $e_{t,i}$ with g_t and f_b^i as in Eq. 4:

$$e_{t,i} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a f_b^i + \mathbf{U}_a g_t), \tag{4}$$

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 Eq. 5:

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

Then we calculate the weighted sum of f_b^i by $\alpha_{t,i}$ to get the attentive global visual features g_b^a of I_b:

$$g_b^a = \sum_{i=1}^{L} \alpha_{t,i} f_b^i, \tag{6}$$

where $g_b^a \in \mathbb{R}^D$. Similarly, we can calculate the attention weights of bottom to top and obtain the attentive global visual features g_t^a of I_t:

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

We then project g_t^a and g_b^a to visual feature vectors v_t^f and $v_b^f \in \mathbb{R}^{m_v}$:

$$v_t^f = \mathrm{ReLU}(\mathbf{W}_p g_t^a), \quad v_b^f = \mathrm{ReLU}(\mathbf{W}_p g_b^a), \tag{8}$$

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

Finally, building on insights from matrix factorization-based methods [22, 23, 24], we also learn top latent factors $T \in \mathbb{R}^{N_T \times m_v}$ and bottom latent factors $B \in \mathbb{R}^{N_b \times m_v}$ 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 v_t^T and v_b^B:

$$v_t^T = T(:, t), \quad v_b^B = B(:, b), \tag{9}$$

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

$$v_t = [v_t^f, v_t^T], \quad v_b = [v_b^f, v_b^B], \tag{10}$$

where v_t and $v_b \in \mathbb{R}^m$, $m = 2m_v$.

3.3 Matching decoder

As shown in Fig. 3(b), we employ a multi-layer neural network to calculate the matching probability of t and b. Given latent representations v_t and v_b calculated in Eq. 10, we first map v_t and v_b into a shared space:

$$h_r = \mathrm{ReLU}(\mathbf{W}_s v_t + \mathbf{U}_s v_b), \tag{11}$$

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

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

where $\mathbf{W}_r \in \mathbb{R}^{2 \times n}$. 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

A shown in Fig. 3(c), we employ a GRU with cross-modality attention as the generation decoder. First, we compute the initial state for the generation decoder with v_t and v_b in Eq. 13:

$$s_0 = \tanh(\mathbf{W}_i v_t + \mathbf{U}_i v_b), \tag{13}$$

where $s_0 \in \mathbb{R}^q$, $\mathbf{W}_i$ 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 $w_{\tau-1}$ and the previous context vector $ctx_{\tau-1}$ as input to compute the new hidden state s_τ and the current output o_τ in Eq. 14:

$$s_\tau, o_\tau = \mathrm{GRU}(w_{\tau-1}, ctx_{\tau-1}, s_{\tau-1}), \tag{14}$$

where $w_{\tau-1} \in \mathbb{R}^e$, $ctx_{\tau-1} \in \mathbb{R}^D$, s_τ and $o_\tau \in \mathbb{R}^q$, and e is the word embedding size. The context vector ctx_τ for the current timestamp τ is computed through the cross-modality attention. It matches the previous state $s_{\tau-1}$ with each element of F_t and F_b to get an importance score. Recall that $F_t = [f_t^1, \dots, f_t^L]$ and $F_b = [f_b^1, \dots, f_b^L]$, we put them together as follows:

$$F_{tb} = [f_{tb}^1, \dots, f_{tb}^{2L}], \quad f_{tb}^i \in \mathbb{R}^D. \tag{15}$$

The context vector ctx_τ is then computed as follows:

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

where $\mathbf{W}_g \in \mathbb{R}^{q \times D}$. The cross-modality attention allows the generation decoder to generate comments with respect to the visual features. o_τ and ctx_τ are used to predict the τ-th word in Eq. 17:

$$p(w_\tau | w_1, w_2, \dots, w_{\tau-1}) = \mathrm{softmax}(\mathbf{W}_o o_\tau + \mathbf{U}_o ctx_\tau), \tag{17}$$

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

3.5 Multi-task learning framework

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

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

where $\mathcal{P}^+ = \{(t_{i_1}, b_{j_1}), (t_{i_2}, b_{j_2}), \dots, (t_{i_N}, b_{j_N})\}, t_i \in \mathcal{T}, b_j \in \mathcal{B}$ is the set of positive combinations, which are top-bottom pairs extracted from the outfit combinations on Polyvore. $\mathcal{P}^- = \{(t, b) \mid t \in \mathcal{T}, b \in \mathcal{B} \wedge (t, b) \notin \mathcal{P}^+\}$ is the set of negative combinations, which are formed by tops and bottoms sampled randomly.

As for the generation task, the loss function is defined in Eq. 19:

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

where $\mathcal{C}^{tb} = \{c_1^{tb}, c_2^{tb}, \dots, c_{N_{tb}}^{tb}\}$ 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:

$$L_{reg} = \|\Theta\|_2^2, \tag{20}$$

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}:

$$L = L_{mat} + L_{gen} + \lambda_{reg} L_{reg}, \tag{21}$$

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* [25], *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, 26, 27]. 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. In addition, we employ *FashionVC* to evaluate the recommendation part.

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

TABLE 1: Dataset statistics.

Dataset	Tops	Bottoms	Outfits	Comments
WoW [2]	17,890	15,996	24,417	–
Exact Street2Shop [25]	–	–	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. 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. The positive bottoms are those that have been matched with a top on Polyvore. The same is true for both top recommendation and bottom recommendation. For the generation task, we evaluate the generated comments based on the the actual user comments for the corresponding outfits. The statistics of *ExpFashion* are listed in Table 1; for comparison we also list the characteristics of 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]

4.2 Implementation details

For the networks in the top and bottom 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 $D = D^1 + D^2 = 64$. And the latent representations size m is set to 600. For the matching decoder, we set the shared space size n to 256. The input and output 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. We set the regularization weight $\lambda_{reg} = 0.0001$. During training, we initialize model parameters randomly using the Xavier method [28]. We use Adam [29] as our optimizing 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 [30] 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 [31] 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 [32]; 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 Tops and Bottoms Recommendation

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

5.1 Methods used for comparison

We consider the following baselines in the tops and bottoms 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 has been paired with the bottom. POP is frequently used as a baseline in recommender systems [33].
- *NRT*: NRT [17] introduces recurrent neural networks into collaborative filtering. As a result, 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 fashion 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 is no reviews in our dataset, we remove the relative part from NRT. Other configurations follow the original paper. In this paper, we not only compare their recommendation performance, but also compare the quality of the generated comments, see Section 6.1.
- *IBR*: IBR [8] aims to model 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.
- *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.

5.2 Evaluation metrics

We employ three evaluation metrics in the tops and bottoms recommendation experiments: *Mean Average Precision* (MAP), *Mean Reciprocal Rank* (MRR), and *Area Under the*

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

ROC curve (AUC). All are widely used evaluation metrics in recommender systems [34, 35, 36].

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

$$\text{MAP} = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{1}{rel_i}\sum_{j=1}^{|B|}(P(j) \times rel(j)), \tag{22}$$

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.

$$\text{MRR} = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{1}{rank_i}, \tag{23}$$

where $rank_i$ refers to the rank position of the first positive bottom for the i-th top.

$$\text{AUC} = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{1}{|E(i)|}\sum_{(j,k)\in E(i)}\delta(f(t_i,b_j) > f(t_i,b_k)) \tag{24}$$

where T is the top collection as queries; $E(i)$ is the set of all positive and negative candidate bottoms for the i-th 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 fashion recommendation results of NFR and the competing models on the ExpFashion dataset are given in Table 2. NFR 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.

TABLE 2: Results of tops and bottoms recommendation on the ExpFashion dataset (%).

Method	Tops			Bottoms		
	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
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
NFR	**11.54**†	**15.38**†	**64.75**†	**13.48**†	**20.83**†	**65.09**†

The superscript † indicates that NFR significantly outperforms BPR-DAE.

(1) NFR significantly outperforms all baselines; NFR achieves the best result on all metrics. Although IBR and BPR-DAE employ pre-trained CNNs (both AlexNet [37] 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 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. We incorporate a 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, NFR can utilize the information of user comments to improve the performance of fashion recommendation. In fact, visual features and user comments are two modalities to explain why a top and a bottom match. NFR 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.

(2) IBR and BPR-DAE both use 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 using an autoencoder neural network.

(3) 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 [38, 39].

(4) The performance of POP is the worst; the reason is that popularity cannot be used to explain why a top and a bottom are matching. In fashion recommendation, the visual feature plays a more decisive role. Incorporating visual signals directly into the recommendation objective can make recommendation more accurate [40]. Because they all use CNNs to extract visual features, IBR, BRP-DAE, and NFR all outperform POP and NRT.

(5) All methods' bottoms recommendation are better than their tops recommendation. This is because in our dataset the average number of positive combinations that each bottom has is more than the average number of positive combinations that each top has. This makes bottoms recommendation easier than tops recommendation.

5.4 Results on FashionVC dataset

In order to confirm the effectiveness of our recommendation part, we also compare NFR-CG, which is NFR without the comment generation part, with POP, IBR and BPR-DAE on the FashionVC dataset; see Table 3. Because there are no comments on FashionVC, we leave out NRT.

TABLE 3: Results of tops and bottoms recommendation on the FashionVC dataset (%).

Method	Tops			Bottoms		
	MAP	MRR	AUC	MAP	MRR	AUC
POP	4.61	5.50	30.10	3.83	4.62	27.13
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
NFR-CG	**8.50**	9.12	**64.17**†	**9.40**	**10.24**	**65.28**†

NFR-CG denotes NFR without the comment generation part. The superscript † indicates that NFR-CG significantly outperforms BPR-DAE.

From Table 3, we can see that NFR-CG achieves the best performance in terms of the MAP and AUC scores on the top recommendation task and also in terms of the MAP, MRR and AUC score on the bottom recommendation task. NFR is only slightly inferior to BPR-DAE in terms of MRR on top recommendation. This means that, even without the generation component, NFR-CG can still achieve better performance than other methods. Our top and bottom encoder

[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
NFR	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 NFR performs significantly better than NRT as given by the 95% confidence interval in the official ROUGE script.

with mutual attention can extract effective visual features for fashion recommendation.

Note that only the differences in terms of AUC are significant. The reason is that the size of FashionVC is small. Although NFR-CG achieves 1.37% and 1.29% increase in terms of MAP and MRR respectively, it is hard to pass the paired t-test with 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 fashion recommendation is able to generate abstractive comments. In order to evaluate the performance of NFR and conduct comparisons against meaningful baselines, we refine existing methods to make them capable of generating comments as follows.

- *LexRank*: LexRank [41] 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 [42] 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 [43] 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 [44] 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 [17] 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:

$$ROUGE\text{-}N_{recall} = \sum_{g_n \in \tilde{c}} \frac{C_{co}(g_n)}{\sum_{g_n \in c} C(g_n)}, \tag{25}$$

where $\tilde{c}$ is the generated comment; c is the ground truth comment; g_n is 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 [45] as another evaluation metric, which is defined as follows:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{26}$$

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

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{1-r/c}, & \text{if } c \leq r, \end{cases} \tag{27}$$

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 comparative methods on the comment generation task are given in Table 4. In order to capture more details, 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 Table 4, we have three main observations:

(1) NFR achieves good performance on the ExpFashion dataset. Especially in terms of BLEU and F-measure of ROUGE-1, ROUGE-L and ROUGE-SU4, NFR gets the best results. NFR is not a top performer on all metrics; for example, LexRank has better performance than NFR in terms of

[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

ROUGE precision. Also RMR's ROUGE recall is better than NFR. This is because LexRank prefers short sentences while RMR prefers long sentences. In contrast, NFR gets much better ROUGE F-measure and BLEU, which means NFR can generate more appropriate comments. In other words, NFR achieves more solid overall performance than other models. The reasons are two-fold. On the one hand, NFR has a top and bottom encoder to encode information of visual features into the latent representations of tops and bottoms. So it makes the latent representations in NFR 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.

(2) One exception to the strong performance of NFR described above is that NFR performs relatively poorly in ROUGE-2. The possible reasons are: (1) The user comments in our dataset are very short, only about 7 words 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 NFR suffers and the superiority of extraction-based methods is clear.

(3) Due to the sparsity of the dataset, NRT performs poorly on most metrics.

7 Analysis and Case Study

TABLE 5: Analysis of attention mechanisms on tops and bottoms recommendation (%).

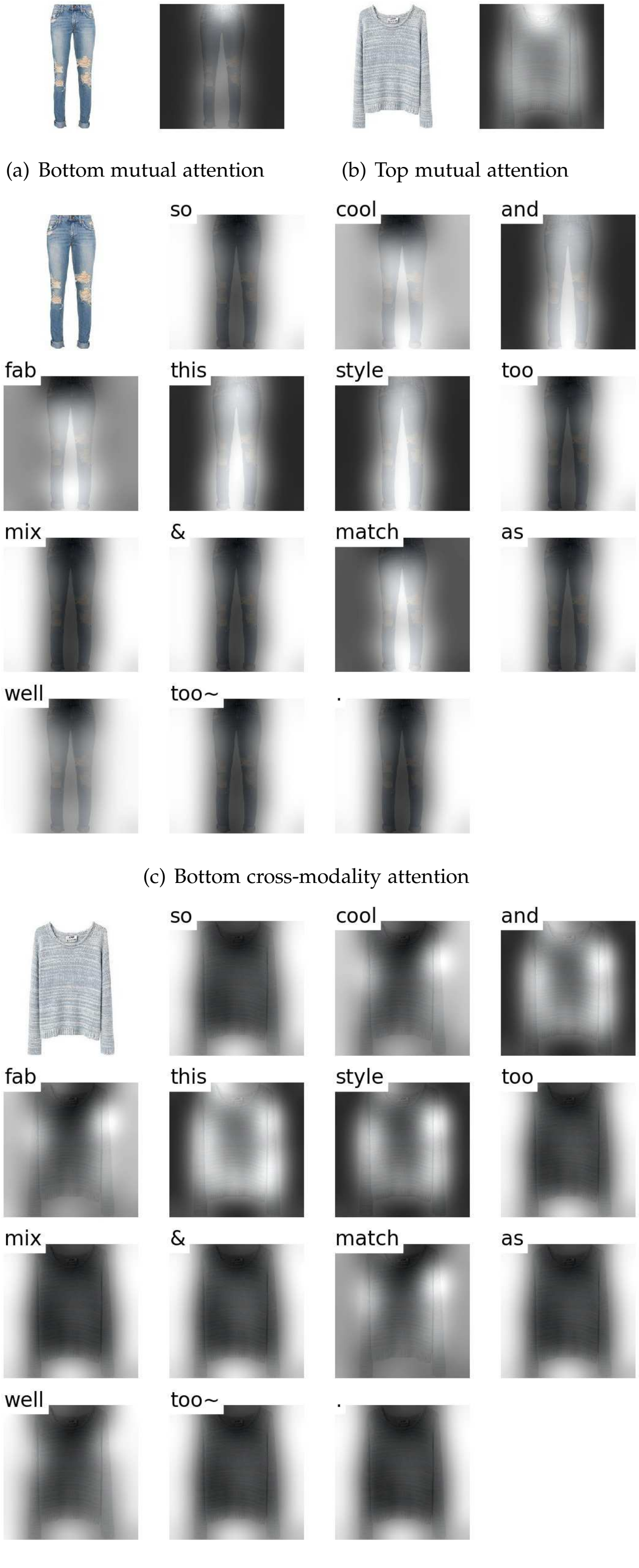
Attention	Tops			Bottoms		
	MAP	MRR	AUC	MAP	MRR	AUC
NFR-NO	10.96	14.93	64.72	13.38	20.13	65.40
NFR-MA	**12.55**	**16.98**	**67.13**	**14.65**	**21.58**	**66.98**
NFR-CA	11.72	15.48	64.85	13.53	20.71	65.60
NFR	11.54	15.38	64.75	13.48	20.83	65.09

NFR-NO: NFR without any attention. NFR-MA: NFR with mutual attention only. NFR-CA: NFR with cross-modality attention only. NFR: NFR has mutual attention and cross-modality attention.

In this section, we conduct further experiments to understand the effectiveness of attentions and multi-task learning, 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 tops and bottoms recommendation and comment generation tasks, we conduct experiments with different settings of NFR. The experimental results are shown in Table 5 and Table 6.



(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.

From Table 5, we notice that NFR-MA (mutual attention only) performs better than NFR-NO (no attention), not only on tops recommendation, but also on bottoms recommendation. We conclude that the mutual attention mechanism can improve the performance of fashion recommendation. Similarly, as shown in Table 6, we observe that NFR-CA (cross-modality attention) outperforms NFR-NO. Thus we conclude that the cross-modality attention mechanism is helpful for the comment generation task. In Table 5, by comparing NFR-MA with NFR, we also find that NFR-

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	
NFR-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
NFR-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
NFR-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
NFR	**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	Tops			Bottoms		
	MAP	MRR	AUC	MAP	MRR	AUC
NFR-CG	10.19	13.65	62.03	12.59	19.17	63.07
NFR	**11.54**	**15.38**	**64.75**	**13.48**	**20.83**	**65.09**

MA outperforms NFR on fashion recommendation. That may be because the two kinds of attention mechanisms can influence each other through joint training. So in NFR the mutual attention mechanism does not reach the same performance as NFR-MA. We think that this performance trade-off is worth making. NFR improves over NFR-MA on comment generation in Table 6. Also, NFR performs better than NFR-NO on both fashion recommendation and comment generation in Table 5 and Table 6, which means that the combination of the two attention mechanisms is effective.

We visualize the effects of both attention mechanisms [46], as shown in Fig. 4. For bottom mutual attention, the waistband and the holes in the pants get more attention. And for top mutual attention, NFR pays more attention to the neckline and the sleeves. When generating comments, NFR also pays different attention at different parts of the top and the bottom. For example, when generating "cool," NFR pays most attention at the bottom, probably because of the holes. And when generating "this style," both the top and the bottom get the main attention. However, for "too" or "mix" which are irrelevant to fashion items, NFR pays little attention to the top and the bottom. So by visualizing attention, we can see that NFR 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 NFR can use user comments to improve the quality of fashion recommendation by multi-task learning, we compare NFR with NFR-CG; see Table 7. We can see that NFR achieves significant improvements over NFR-CG; on tops recommendation, MAP increases by 1.35%, MRR increases by 1.73%, AUC increases by 2.72%, and on bottoms recommendation, MAP increases by 0.89%, MRR increases by 1.66%, AUC increases by 2.02%. By jointly learning, our multi-task framework NFR learns shared representations [47] for both recommendation and generation, which can make effective use of the information in comments to improve recommendation performance.

Additionally, through comparing NFR-CG with POP, IBR and BPR-DAE, we find that, on ExpFashion, NFR-CG also achieves comparable results to other methods, which is consistent with the results on FashionVC (see Section 5.4).

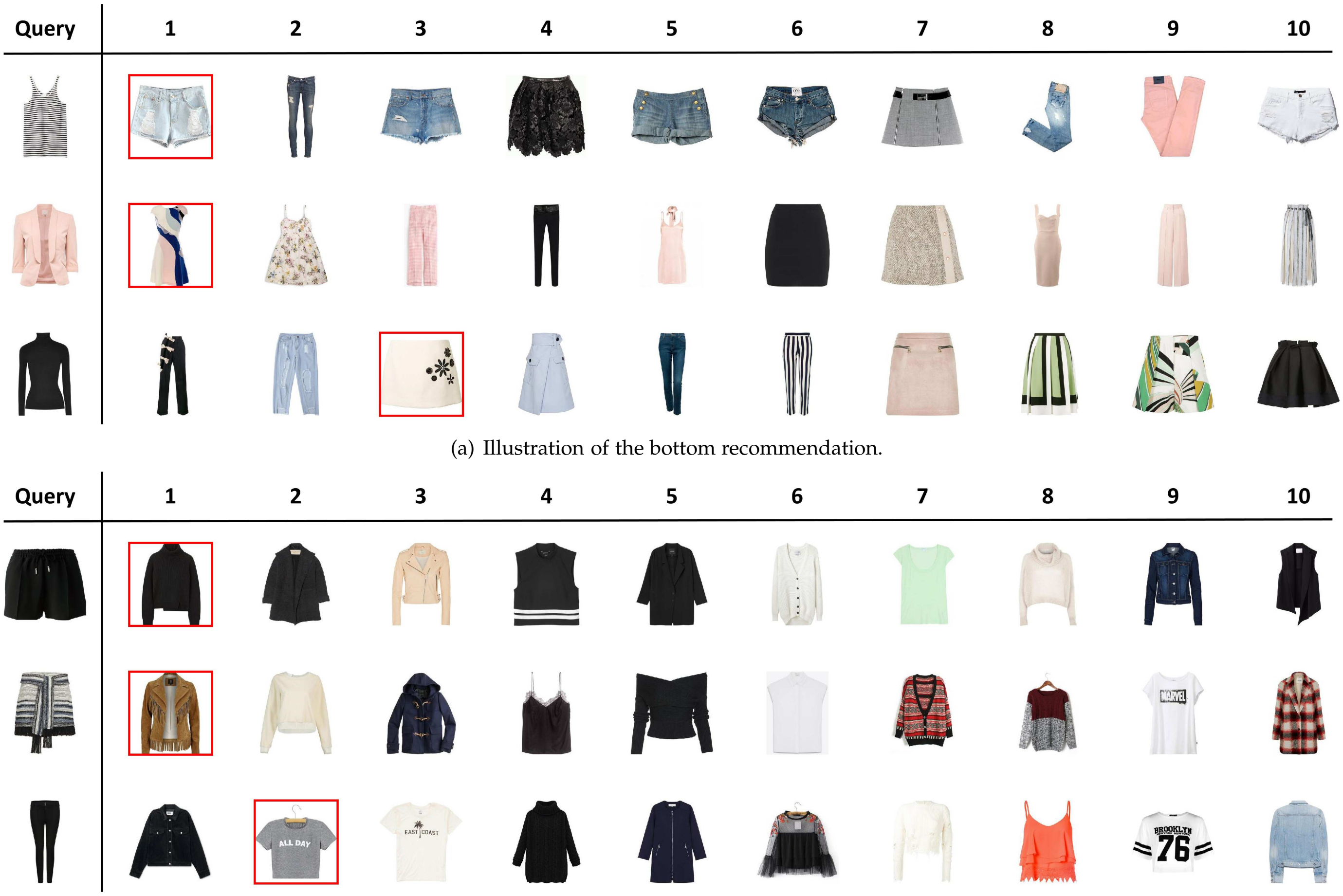
7.3 Recommendation case studies

In Fig. 5 we list some recommendation results of NFR 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 NFR 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 NFR can indeed provide good recommendations.

7.4 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 ExpFashion dataset, shown in Table 8. 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, for "wow! this is so beautiful! love the skirt!", we think that "this is so beautiful" shows love and appreciation to this combination, which expresses the feelings reflected in most user comments, and "love the skirt" expresses a special preference for the skirt, which is also a affirmation to this top-bottom pair. "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, NFR 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, NFR 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 for other users' comments, not a comment posted for the

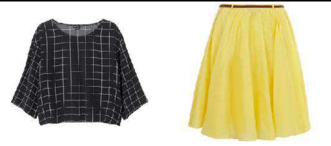


(a) Illustration of the bottom recommendation.

(b) Illustration of the top recommendation.

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

TABLE 8: 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 ! (✗)

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 NFR 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 fashion 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 NFR, which simultaneously gives fashion 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 NFR and have found significant improvements over state-of-the-art baselines in terms of MAP, MRR and AUC. Moreover, we have found that NFR 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 fashion recommendation and comment generation.

Limitations of our work include the fact that NFR 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, NFR 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 [48] or dialogue systems can be explored [49, 50]. Finally, we would like to incorporate other mechanisms, such as auto-encoder, to further improve the performance.

Acknowledgments

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, Amsterdam Data Science, the Bloomberg Research Grant program, Elsevier, the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Google Faculty Research Awards program, the Microsoft Research Ph.D. program, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs CI-14-25, 652.002.001, 612.001.551, 652.-001.003, and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

[1] Tomoharu Iwata, Shinji Watanabe, and Hiroshi Sawada, “Fashion coordinates recommender system using photographs from fashion magazines,” in *International Joint Conference on Artificial Intelligence*, 2011, pp. 2262–2267.

[2] Si Liu, Jiashi Feng, Zheng Song, Tianzhu Zhang, Hanqing Lu, Changsheng Xu, and Shuicheng Yan, “Hi, magic closet, tell me what to wear!” in *ACM Multimedia*, 2012, pp. 619–628.

[3] Nava Tintarev and Judith Masthoff, “A survey of explanations in recommender systems,” in *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*. IEEE, 2007, pp. 801–810.

[4] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma, “Neurostylist: Neural compatibility modeling for clothing matching,” in *ACM Multimedia*, 2017, pp. 753–761.

[5] Yang Hu, Xi Yi, and Larry S. Davis, “Collaborative fashion recommendation: A functional tensor factorization approach,” in *ACM Multimedia*, 2015, pp. 129–138.

[6] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, “Bpr: Bayesian personalized ranking from implicit feedback,” in *Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.

[7] Vignesh Jagadeesh, Robinson Piramuthu, Anurag Bhardwaj, Wei Di, and Neel Sundaresan, “Large scale visual recommendations from street fashion images,” in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2014, pp. 1925–1934.

[8] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel, “Image-based recommendations on styles and substitutes,” in *International Conference on Research and Development in Information Retrieval*, 2015, pp. 43–52.

[9] Ruining He and Julian McAuley, “VBPR: Visual bayesian personalized ranking from implicit feedback,” in *AAAI Conference on Artificial Intelligence*, 2016, pp. 144–150.

[10] Yuncheng Li, Liangliang Cao, Jiang Zhu, and Jiebo Luo, “Mining fashion outfit composition using an end-to-end deep learning approach on set data,” in *IEEE Transactions on Multimedia*, vol. 19. IEEE, 2017, pp. 1946–1955.

[11] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S. Davis, “Learning fashion compatibility with bidirectional lstms,” in *ACM Multimedia*, 2017, pp. 1078–1086.

[12] Jesse Vig, Shilad Sen, and John Riedl, “Tagsplanations: Explaining recommendations using tags,” in *International Conference on Intelligent User Interfaces*, 2009, pp. 47–56.

[13] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma, “Explicit factor models for explainable recommendation based on phrase-level

sentiment analysis," in *International Conference on Research and Development in Information Retrieval*, 2014, pp. 83–92.

[14] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen, "Trirank: Review-aware explainable recommendation by modeling aspects," in *ACM International Conference on Information and Knowledge Management*, 2015, pp. 1661–1670.

[15] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, ""Why should I trust you?": Explaining the predictions of any classifier," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[16] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke, "Social collaborative viewpoint regression with explainable recommendations," in *International Conference on Web Search and Data Mining*, 2017, pp. 485–494.

[17] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam, "Neural rating regression with abstractive tips generation for recommendation," in *International Conference on Research and Development in Information Retrieval*, 2017, pp. 345–354.

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11. IEEE, 1998, pp. 2278–2324.

[19] Kyunghyun Cho, Bart Van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014, pp. 103–111.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[21] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[22] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix factorization techniques for recommender systems," in *IEEE Computer Society Press*, vol. 42, no. 8. IEEE, 2009, pp. 30–37.

[23] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Annual Conference on Neural Information Processing Systems*, 2000, pp. 535–541.

[24] Ruslan Salakhutdinov and Andriy Mnih, "Probabilistic matrix factorization," in *Annual Conference on Neural Information Processing Systems*, 2007, pp. 1257–1264.

[25] M. Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C. Berg, and Tamara L. Berg, "Where to buy it: Matching street clothing photos in online shops," in *IEEE International Conference on Computer Vision*, 2015, pp. 3343–3351.

[26] Kota Yamaguchi, M. Hadi Kiapour, Luis E.Ortiz, and Tamara L. Berg, "Parsing clothing in fashion photographs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3570–3577.

[27] Kota Yamaguchi, M. Hadi Kiapour, Luis E. Ortiz, and Tamara L. Berg, "Retrieving similar styles to parse clothing," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 5. IEEE, 2015, pp. 1028–1040.

[28] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Journal of Machine Learning Research*, vol. 9, 2010, pp. 249–256.

[29] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[30] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. III–1310–III–1318.

[31] Philipp Koehn, "Pharaoh: A beam search decoder for phrase-based statistical machine translation models," in *Association for Machine Translation in the Americas*, 2004, pp. 115–124.

[32] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," in *CoRR*, vol. abs/1603.04467, 2015.

[33] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua, "Neural collaborative filtering," in *International World Wide Web Conference*, 2017, pp. 173–182.

[34] Steffen Rendle and Lars Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *International Conference on Web Search and Data Mining*, 2010, pp. 81–90.

[35] Hanwang Zhang, Zheng-Jun Zha, Yang Yang, Shuicheng Yan, Yue Gao, and Tat-Seng Chua, "Attribute-augmented semantic hierarchy: Towards bridging semantic gap and intention gap in image retrieval," in *ACM Multimedia*, 2013, pp. 33–42.

[36] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma, "Neural attentive session-based recommendation," in *ACM International Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.

[37] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM Multimedia*. ACM, 2014, pp. 675–678.

[38] Xueming Qian, He Feng, Guoshuai Zhao, and Tao Mei, "Personalized recommendation combining user interest and social circle," in *IEEE Transactions on Knowledge & Data Engineering*, vol. 26, no. 7, 2014, pp. 1763–1777.

[39] Xinxi Wang and Ye Wang, "Improving content-based and hybrid music recommendation using deep learning," in *ACM Multimedia*, 2014, pp. 627–636.

[40] Wang Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley, "Visually-aware fashion recommendation and design with generative image models," in *Industrial Conference on Data Mining*, 2017.

[41] Günes Erkan and Dragomir R. Radev, "Lexrank:

Graph-based lexical centrality as salience in text summarization," in *Journal of Artificial Intelligence Research*, vol. 22, no. 1. AI Access Foundation, 2004, pp. 457–479.

[42] Chong Wang and David M. Blei, "Collaborative topic modeling for recommending scientific articles," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2011, pp. 448–456.

[43] Guang Ling, Michael R. Lyu, and Irwin King, "Ratings meet reviews, a combined approach to recommend," in *ACM RecSys*, 2014, pp. 105–112.

[44] Chin-Yew Lin, "Rouge: a package for automatic evaluation of summaries," in *Workshop on Text Summarization Branches Out, The Association for Computational Linguistics*, 2004.

[45] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: A method for automatic evaluation of machine translation," in *The Association for Computational Linguistics*, 2002, pp. 311–318.

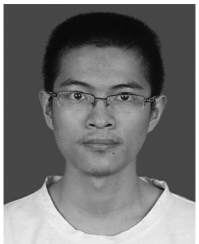
[46] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[47] Rich Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

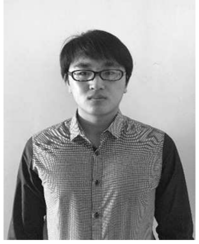
[48] Jiyin He, Wouter Weerkamp, Martha Larson, and Maarten de Rijke, "An effective coherence measure to determine topical consistency in user generated content," *International Journal on Document Analysis and Recognition*, vol. 12, no. 3, pp. 185–203, September 2009.

[49] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan, "A diversity-promoting objective function for neural conversation models," in *The North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 110–119.

[50] Svitlana Vakulenko, Maarten de Rijke, Michael Cochez, Vadim Savenkov, and Axel Polleres, "Measuring semantic coherence of a conversation," in *ISWC 2018: 17th International Semantic Web Conference*. Springer, October 2018.



Yujie Lin received B.S. from Shandong University, in 2016. Currently he is a master in Shandong University, supervised by Jun Ma. His research area is in information retrieval, recommender system and deep learning.



Pengjie Ren is a Ph.D. student in Information Retrieval Lab at Shandong University, supervised by Prof. Jun Ma. His research interests fall in information retrieval, natural language processing, and recommender systems. He has previously published at TOIS, SIGIR, CIKM, and COLING.

Zhumin Chen is an associate professor in 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. He received his Ph.D. from Shandong University. His research interests mainly include information retrieval, big data mining and processing, as well as social media processing.

Zhaochun Ren received his MSc degree from Shandong University in 2012, and the PhD degree from University of Amsterdam in 2016. He is a research scientist in JD.com. He previously worked as a research associate in University of London. He also worked as a short-term visiting scholar in Max-Planck-Institut fr 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 his BSc, MSc, and PhD degrees from Shandong University in China, Ibaraki University, and Kyushu University in Japan, respectively. He is a full professor at Shandong University. He was a senior researcher in the Department of Computer Science at Ibaraki Univsity in 1994 and German GMD and Fraunhofer from the year 1999 to 2003. His research interests include information retrieval, Web data mining, recommendation systems and machine learning. He has published more than 150 International Journal and conference papers, including SIGIR, MM, TOIS and TKDE. He is a member of the ACM and IEEE.

Maarten de Rijke received the MSc degrees in philosophy and mathematics, and the PhD degree in theoretical computer science. He is a professor in computer science in the Informatics Institute, 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 and of the Springer's Information Retrieval book series.