PAPER

Cross-Domain Outfit Recommendation via Bidirectional LSTM and Deep Metric Learning

Hiroshi MATSUI[†], ***Sophia University***

SUMMARY *Outfit recommendation systems solve the problem of deciding which garment to wear or buy. Image-based outfit recommendation systems receive garment images from users, and then recommend an outfit (a set of visually compatible garments with the inputs). It is easier for users to prepare street images (garment images taken by themselves) than shop images (garment images used in online shopping websites). However, in previous studies on image-based outfit recommendations, there have been few methods appropriate to handle street image inputs due to the gap between street and shop image features. To address the feature gap issue, we have adapted the previously proposed recommendation model using bidirectional LSTM to handle street image inputs, using the deep metric learning. Our approach has outperformed the previous model in four tasks including two novel tasks for outfit recommendations with street inputs.*

1. INTRODUCTION

Outfit recommendation systems solve the problem of deciding which garment to wear or buy. Image-based outfit recommendation systems receive garment images from users, and then recommend an outfit (a set of visually compatible garments with the inputs).

For image-based outfit recommendations, the main approach is to learn the visual compatibility among garments. There have been some studies on the *pair-wise* compatibility [9], [13], [3], which is the relation between only two garments (*e.g.,* a white blouse is more compatible with a floral skirt than a floral T-shirt). In order to capture the compatibility, they trained a convolutional neural network (CNN) as a feature extractor of garment images, using the metric learning. More specifically, in the metric space, the compatible garment features must be close to each other. Compared to them, Han *et al,*, (2017) [2] succeeded in capturing the compatibility of an entire outfit. They considered an outfit as a sequence of garment features and trained bi-directional LSTM (Bi-LSTM) as a language model. Their model learns the compatibility as the time dependencies among garment features in a sequence.

Garment images are categorized into two main groups: *shop images* (*e.g.,* garment images used in online shopping websites) and *street images* (*e.g.,* garment images taken by themselves), as shown in Figure 1. It is easier for users to prepare street images than shop images. However, the previous studies only focus on shop image inputs *i.e.,* their models are not appropriate to handle street image inputs due to the feature gap between street and shop images. We define *cross-domain image-based outfit recommendation* as the outfit recommendation in which the compatible shop images are recommended, given street images.

To address the problem, we have proposed Bi-LSTM Street-to-Shop (Bi-LSTM-S2S), which consists of two modules: S2S module and sequence module. The main idea of our approach is to adapt the previously proposed Bi-LSTM-based model [2] to handle street image inputs, using metric learning. The S2S module is a CNN-based image feature extractor. In order to obtain features without the gap, this module is trained, using the metric learning. We define a *positive pair* as a street-shop image pair of the same item, and a *negative pair* as that of different items. Through the metric learning, the feature dot product of positive pairs must be larger than that of negative pairs. Following [2], we employed a Bi-LSTM as the sequence module. This module aims to learn the time dependencies within a garment feature sequence (an outfit) as the compatibility of the outfit.

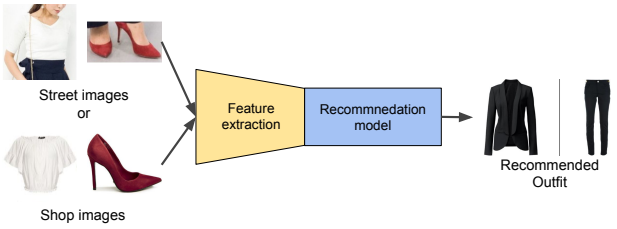


Fig. 1 Example of image-based outfit recommendation.

We evaluated our model on four tasks: (1) cross-domain image retrieval (CDIR), (2) fill-in-the-blank (FITB), (3) fill-in-the-blank street-to-shop (FITB-S2S), and (4) outfit recommendation S2S. The CDIR retrieves a shop image of the same garment as a given street image. The purpose of the CDIR is to confirm that the S2S module can reduce the feature gap. In the FITB task introduced in [2], outfit recommendation models predict the most compatible garment choice, given an incomplete outfit. Both incomplete outfits and choices consist of only shop images. FITB-S2S is a variant of FITB, whose incomplete outfits consist of street images while its choices are shop images, as the FITB. In the outfit recommendation S2S, models recommend an outfit using shop images, given street images.

2. RELATED WORKS

2.1 Fashion Compatibility Learning

We categorize fashion compatibility into two types: (1) *pair-wise compatibility* and (2) *outfit compatibility*, or *fashionability*.

McAuley *et al.,* 2015 [9], Veit *et al.,* 2015 [13], He *et al.,* 2016 [3], studied the pair-wise compatibility. They used the Amazon co-purchase dataset and consider that two garments bought together are compatible. To learn the compatibility, they employed a convolutional neural network (CNN) as the image feature extractor and trained it using the metric learning. In the metric space, the distance between the compatible garments must be smaller than that of the incompatible garments. The models recommend an outfit, iteratively adding the nearest garments to a query garment. These pair-wise approaches could measure the compatibility of an outfit as a whole (outfit compatibility) with some voting strategy using all pairs in the outfit. However, this would incur high computational cost when the outfit is large.

In contrast to them, Han *et al.,* 2017 [2] represents an outfit as a sequence of garment image features extracted by a CNN. They trained bidirectional LSTM (Bi-LSTM) as the language model to capture the compatibility of an entire outfit, considering that garments in the same outfit are compatible. They showed their model outperformed one of the pair-wise approaches on quantitative experiments. In this study, we follow their Bi-LSTM-based framework because of their effectiveness and computational efficiency.

2.2 Cross-Domain Image Retrieval

Cross-domain image retrieval (CDIR) retrieves the corresponding image from a domain, given an image from another domain. More specifically, as a fashion application, the CDIR retrieves a shop image of the same garment as a given street image. There have been some studies on the CDIR in the fashion domain [1], [4], [8], [6]. They tried to obtain garment image features without the gap between the street and shop domains, using the metric learning. In the metric space, the feature distance between the street-shop image pairs with the same item must be smaller than that with different items. Following the previous studies, we obtained such cross-domain features and utilized them for cross-domain outfit recommendations.

3. METHOD

Figure 2 illustrates an overview of our model, which consists of two modules: street-to-shop (S2S) module and sequence module.

3.1 Street-to-Shop Module

In order to obtain features without the gap between street and shop images, we trained the S2S module using metric learning, defining a street-shop image pair containing the same garment as a *positive pair*, and different garments as a *negative pair*. A convolutional neural network (CNN) is employed as an image feature extractor. In a training process, the following hinge loss is minimized so that feature dot products of a positive pair must be larger than those of a negative pair by a margin of m.

$$L_{street} = \frac{1}{B}(\sum_{i=1}^{B}\sum_{j=1,j\neq i}^{B} \max(0, -x_{street}^{(i)T}x_{shop}^{(i)} + x_{street}^{(i)T}x_{shop}^{(j)} + m)) \quad (1)$$

where B denotes a mini-batch size, x_{street} is a street image feature and x_{shop} is a shop image feature, i is the index of a positive pair, and j is the index of a negative sample (a shop image which forms a negative pair with the i-th street image). We define L_{shop}, swapping $street$ and $shop$, and eventually minimize $L_{street} + L_{shop}$.

3.2 Sequence Module

Following [2], a Bi-LSTM is employed as a sequence module to learn fashion compatibility. Bi-LSTM is a variant of LSTM processing an input sequence in both forward and backward directions.

Training Process: This module receives an outfit with T garments as a feature sequence $\{x_1, \ldots, x_T\}$ extracted by the S2S module and then maps them to hidden states $\{h_1, \ldots, h_T\}$, where h_t is the prediction of the next garment feature x_{t+1}. In the training process, this module is trained to maximize the probability of the next garment given the previous garments, treating this probability as the compatibility of the next garment. More formally, we minimize the following loss function:

$$L_f = -\frac{1}{T}\sum_{t=1}^{T} \log P(x_{t+1}|x_1, \ldots, x_t; \theta_f) \quad (2)$$

$$P(x_{t+1}|x_1, \ldots, x_t; \theta_f) = \frac{\exp(h_t^{(f)}x_{t+1})}{\sum_{x\in\mathcal{X}}\exp(h_t^{(f)}x)} \quad (3)$$

where $\mathcal{X}$ denotes all garment features from the current mini-batch (we used 5 as the mini-batch size), $h_t^{(f)}$ is the t-th hidden states of the forward LSTM *i.e.,* the prediction of the next garment feature x_{t+1}, and θ_f is the parameter of the forward LSTM. We also define the backward loss analogously to the forward loss as follows:

$$L_b = -\frac{1}{T}\sum_{t=T}^{1} \log P(x_{t-1}|x_T, \ldots, x_t; \theta_b) \quad (4)$$

$$P(x_{t-1}|x_T, \ldots, x_t; \theta_b) = \frac{\exp(h_t^{(b)}x_{t+1})}{\sum_{x\in\mathcal{X}}\exp(h_t^{(b)}x)} \quad (5)$$

where $h_t^{(b)}$ is the t-th hidden states of the backward LSTM *i.e.,* the prediction of the previous garment feature x_{t-1}, and θ_b is the parameter of the backward LSTM. We eventually minimize $L_f + L_b$. Note that we added zero vectors x_0 and x_{T+1} to each side of the input sequence as the *begining-of-sequence (BOS)* and *end-of-sequence (EOS)*, respectively.

Compatible Garment Prediction: Following [2], in the inference, the sequence module predicts the t-th garment c^* in an outfit, based on the following objective function:

$$c^* = \arg\max_{c \in C} P(x_t = x_c \mid x_1, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T)$$
$$= \arg\max_{c \in C} P(x_t = x_c \mid x_1, \ldots, x_{t-1}; \theta_f)$$
$$P(x_t = x_c \mid x_T, \ldots, x_{t+1}; \theta_b) \quad (6)$$

where c denotes a garment choice in all choices C, c^* denotes the most compatible garment, and x_c denotes the image feature of a choice c. Note that when to predict garments at the side (t = 1 or T), the module uses only the backward probability or the forward probability, respectively. Our model solves the FITB, the FITB-S2S, and the outfit recommendation S2S, based on Eq.6.

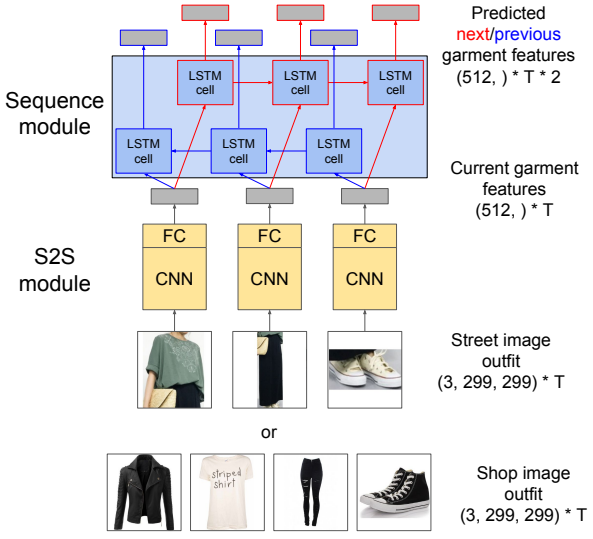


Fig. 2 The overview of our model, Bi-LSTM-S2S

4. EXPERIMENTS

We evaluated our model on four tasks: (1) cross-domain image retrieval (CDIR), (2) fill-in-the-blank (FITB), (3) fill-in-the-blank street-to-shop (FITB-S2S), and (4) outfit recommendation S2S.

4.1 Cross-Domain Image Retrieval

CDIR retrieves a shop image of the same garment as a given street image. We compared the S2S module with baselines to confirm that the S2S module can reduce the feature gap.
Eavluation Metrics: We adopted the top-k accuracy against N images (acc.@k/N) as an evaluation metric. For each sample in a test dataset, a score of 1 or 0 is computed as follows: a model receives a street image and then retrieves the top-k nearest shop images from N shop images. If the nearest shop images contain the same garment as the street image, the score of 1 is given to the sample. If not so, the score of 0 is given. Then, the Acc.@k/N for the model is computed as the mean of the scores over all samples in the test dataset. The higher the Acc.@k/N is, the better a model is. We adopted $k = 20$ and $N = 1,000$ in this study. We specify the number of images N, because Huang, 2015 [4] showed that the top-k accuracy depends on N *e.g.,* the larger N is, the lower the top-k accuracy is.
Dataset: We created a novel cross-domain fashion image dataset, *WEAR*, collecting 800K fashion *snap images* (the photo depicting a person wearing an outfit) from wear.jp and the corresponding shop images from zozo.jp, Figure 3(a) shows an example of the WEAR dataset.

We processed this dataset for CDIR as *WEAR-Pairs*, which contains a large amount of street-shop image pairs as shown in Figure 3(b). Street images in WEAR-Pairs are the garment region detected from a snap image by the single shot multibox detector (SSD) [7] trained on ModaNet dataset [14].

We also used the Polyvore dataset [2] to train the baseline (we will describe it in the next **Baselines**) and to qualitatively evaluate the S2S module. The qualitative evaluation aims to confirm whether the S2S module can retrieve the similar Polyvore shop image, given a WEAR shop image *i.e.,* whether there is no large gap between the features from Polyvore and WEAR. Figure 4 shows example outfits in the Polyvore dataset.
Baselines: Our S2S module was compared with the following alternatives:

- **Bi-LSTM**: the CNN-based feature extractor used in the previous model trained on Polyvore shop image dataset according to [2].
- **ImageNet**: the CNN-based feature extractor trained on ImageNet classification task [5] without any fine-tuning.

Implementation Details: The S2S module was trained on WEAR. For the optimization, we used the Adam optimizer with a batch size of 30 and the initial learning rate of 2×10^{-4}. The weight decay value was 5×10^{-3}. The CNNs used in S2S and Bi-LSTM were pre-trained on ImageNet classification task [5].

For the comparison to the baselines, we trained Hinge+Dot+M187 for 20K iterations, which was the best model in the hyperparameter search.

The following settings were shared in all experiments in this study: Each shop and street image was resized to 299×299, keeping the aspect ratio and filling the blank with white (255) before being fed into a model. We adopted Inception-V3 [11] as the CNN architecture. All models were implemented with Chainer [12] and trained on GeForce GTX 1080 GPU.
Results and Discussion: Table 1 demonstrates the S2S module significantly outperformed the extractor of Bi-LSTM, indicating that the S2S module reduces the feature gap. Some retrieval examples by the methods are illustrated in Figure 5

(a) WEAR (b) WEAR-Pairs

Fig. 3 Examples of WEAR dataset and WEAR-Pairs dataset.

(a) Valid outfit

(b) Outfit with accessories

(c) Outfit with category disorder

(d) Outfit with category duplication

Fig. 4 Example outfits of Polyvore dataset proposed in [2] and its three problems.

Table 1 Comparison between our method and the alternative on the cross-domain image retrieval (CDIR) task.

Method	Acc.@20/1000 (↑)
Bi-LSTM	8.60%
ImageNet	30.0%
S2S (Ours)	**67.1**%

and Figure 6.

We also conducted the CDIR, searching the Polyvore shop images for a similar item to a given WEAR street image. Figure 7 shows an example of the result, in which the top-20 retrievals are similar check skirts to the query. This indicates that there is no large feature gap between the Polyvore shop images and the WEAR street images.

4.2 Fill-in-the-Blank

In the FITB task introduced in [2], outfit recommendation models predict the most compatible garment choice, given an incomplete outfit, using only shop images. Our model predicts the answer using Eq. 6.

Evaluation Metrics: We adopted the accuracy as an evaluation metric according to [2]. For each sample in a test dataset, a score of 1 or 0 is computed as follows: A model receives an incomplete outfit and garment choices, and then

Fig. 5 Top-20 retrieval results of Bi-LSTM and S2S. In each result, a row consists of four sub-rows displaying top-20 retrieved shop images, from left to right, top to bottom. The correct shop images are surrounded by green frame.

Fig. 6 Top-20 retrieval results of Bi-LSTM and S2S. In each result, a row consists of four sub-rows displaying top-20 retrieved shop images, from left to right, top to bottom. The correct shop images are surrounded by green frame.

Fig. 7 Top-20 retrieved Polyvore shop images by S2S, given a WEAR street image. Most left column is a street image query. Four rows at the right side of the query displays the top-20 retrieved shop images, from left to right, top to bottom.

select the answer from the choices. If the answer is correct, the score of 1 is given to the sample. If not so, the score of 0 is given. Then, the accuracy for the model is computed as the mean of the scores over all samples in the test dataset. The higher the accuracy is, the better the model is.

Dataset: We created *Polyvore Category Restricted (Polyvore-CR)* dataset for more practical and simpler outfit recommendations, fixing the three problems with the Polyvore dataset [2]: (1) *accessories*, (2) *category disorder*, and (3) *category duplication*, as shown in Figure 4.

In the first problem, accessories are not necessary to compose an outfit, and thus make outfit recommendations difficult. Therefore, we removed these category items from the Polyvore.

In the second, when feeding an outfit into the sequence module, the category order is not always the same *i.e.,* many outfits are fed by the following order, while some outfits are not - *tops (shirts/t-shirts, outerwears), bottoms, shoes, and accessories (handbags, hats, glasses, watches, necklaces, earrings, etc*). Such coarse categories don't exist in the dataset while it has only fine-grained categories (*e.g., cardigans, rompers, knee-length skirts, boots, etc*). Thus we categorized the fine-grained categories into the following new coarse categories and then fed the outfits by the order - *outerwears, tops, full (dresses/one-pieces etc), bottoms, shoes*.

In the last, some outfits have the same category garments(*e.g.,* an outfit consisting of four skirts). Since we cannot wear these outfits in practice, we define them as *invalid outfits*. Thus, we removed these category-duplicated invalid outfits, excluding the *tops* duplication, which is usual (*e.g.,* we can wear a sweater and a shirt together). We kept only the outermost top in an outfit (*e.g.,* if an outfit contains a sweater and a shirt, we only keep the sweater and remove the shirt) according the outfits in the WEAR. The street image outfits in the WEAR have only the outermost top because they are detected from the snap images.

We also processed the Polyvore-CR as *Polyvore-CR-FITB* for the FITB evaluation. Following [2], for each outfit, we randomly selected one garment (the *correct garment*) and replaced it with a blank, and then randomly select some garments (the *wrong* garments) from other outfits. We assume that the randomly selected garments are less compatible than the correct one. Models predict the correct one from the choices (the correct one and the wrong garments). Unlike [2], we unified the coarse category of the choices, while their fine-grained categories must be different *e.g.,* if the correct garment is a *denim jacket* (a kind of outerwears), we select a *coat* (a kind of outerwears) rather than a pair of *sneakers* (a kind of shoes). That's because, without the unification, FITB could have been just a trivial task to predict not the compatible garment but the compatible *category*. This easy task is not practical since no one can't decide which to wear as their outerwear, a denim jacket or a pair of sneakers. This unification makes the task more difficult, increasing hard negatives (the wrong garments as compatible or more than the correct garment) in the choices. Thus, we selected the wrong garments from the different fine-grained categories and increased the number of choices from 4 to 10 against [2].

Baselines: Our method was compared with the following alternatives:

- **Bi-LSTM** [2]: This consists of a CNN-based feature extractor module and the same Bi-LSTM-based sequence module as ours. In contrast to ours, these modules were jointly trained. This method solve the task by the same way as our method, using Eq.6.
- **random guessing**: Randomly choosing a garment from choices. Since the number of choices is 10, its accuracy is 10%.

Implementation Details: For Bi-LSTM-S2S (ours), we used the same S2S module pre-trained in the CDIR and fixed the weights while training its sequence module.

Both Bi-LSTM-based models shared the following configurations: both models were trained on the Polyvore-CR dataset with the batch size of 5 and the 10K iterations. The other training configurations are the same as [2].

Results and Discussion: As shown in the second column of Table 2, Bi-LSTM-S2S outperformed the random guessing, though it did not outperform Bi-LSTM. This indicates that our model recommends a more compatible garment than the random guessing, given an incomplete shop image outfit.

Table 2 Comparison between our method and alternatives on FITB and FITB-S2S. The FITB-S2S was conducted on three training dataset: the Polyvore-CR (Polyvore), the WEAR-BboxSeq (WEAR), and the mixture of them (Poly+WEAR)

Method	Acc. (↑)			
	FITB	FITB-S2S		
		Polyvore	WEAR	Poly+WEAR
random	10.0%	10.0%	10.0%	10.0%
Bi-LSTM	**25.2**%	10.9%	12.4%	11.2%
Bi-LSTM-S2S (Ours)	16.3%	**14.5**%	**20.6**%	**17.9**%

4.3 Fill-in-the-Blank Street-to-Shop

We newly introduced a variant of FITB, FITB-S2S, whose incomplete outfits consist of street images and choices consists of shop images. We used the same evaluation metric and compared the same methods as the FITB.

Dataset: We created two datasets from the WEAR dataset: *WEAR Bounding box Sequences (WEAR-BboxSeq)* for the training, and *WEAR Fill-in-the-Blank Street-to-Shop (WEAR-FITB-S2S)* for the evaluation. WEAR-BboxSeq consists of a huge amount of the street image sequences detected from a snap image in WEAR. WEAR-FITB-S2S is created by the analogous way to make the Polyvore-CR-FITB except its incomplete outfits consist of the WEAR street images instead of shop images.

We conducted this experiment on three training dataset: Polyvore-CR (shop images), WEAR-BboxSeq (street images), and the 1:1 mixture of them. In all training dataset, we evaluated the models on only WEAR-FITB-S2S.

Result and Discussion: As shown in Table 2, in both training data, our approach showed the best performance in all methods, indicating that Bi-LSTM-S2S can recommend a more compatible shop image than Bi-LSTM, given an incomplete street image outfit. For all training dataset, our model also outperformed the Bi-LSTM trained on the mixture dataset (Poly+WEAR), indicating that our metric learning approach is more effective for the FITB than just training on the mixture of street and shop image dataset. As shown in Figure 8, our model was able to predict the most compatible garment, while the Bi-LSTM failed with higher confidence. Furthermore, the similar garment choices have the close compatibility score for our model, though they do not for the Bi-LSTM, *e.g.,* as shown in Figure 8(b), the three pairs of *black heels* are at the left side together in our result, though they are not put together in that of the Bi-LSTM. This indicates that our Bi-LSTM-S2S can correctly recognize street and shop images, while the Bi-LSTM cannot.

4.4 Outfit Recommendation Street-to-Shop

We also introduced a new task, outfit recommendation S2S, where models recommend an outfit using shop images, given street images. Our model predicts the outside compatible garments using Eq.6, and append or prepend the predicted garments to the outfit step by step in both directions, as shown in Figure 9. Following [10], we also adopted the w-width beam search as shown in Figure 10.

Baselines Our model was compared with the Bi-LSTM [2]. This model recommended outfits in the same way as ours.

Implementation Details We used the same models trained on the WEAR-BboxSeq as the FITB-S2S and adopted $w = 3$ for the width of the beam search.

Results and Discussion: Figure 11 shows recommended outfits, given a street image input. For the left beige blouse query, our model recommended compatible outfits though the Bi-LSTM seemed to recommend a pair of blue-toe shoes

(a) Result of the models trained on Polyvore dataset.

(b) Result of the models traiend on WEAR dataset.

Fig. 8 Results of the previously proposed Bi-LSTM and our Bi-LSTM-S2S on fill-in-the-blank street-to-shop (FTIB-S2S). Each row consists of shop image choices sorted by their predicted compatibility from left to right, from top to bottom *i.e.,* the top left one in each row is the most compatible choice for the blank in the incomplete outfit, while the bottom right one is the worst. The number under each image is its compatibility ranged in $[0, 1]$. Bright colors mean high compatibilities. The correct shop image is surrounded by a red frame.

as the bottom, indicating that the Bi-LSTM cannot correctly recognize shop images.

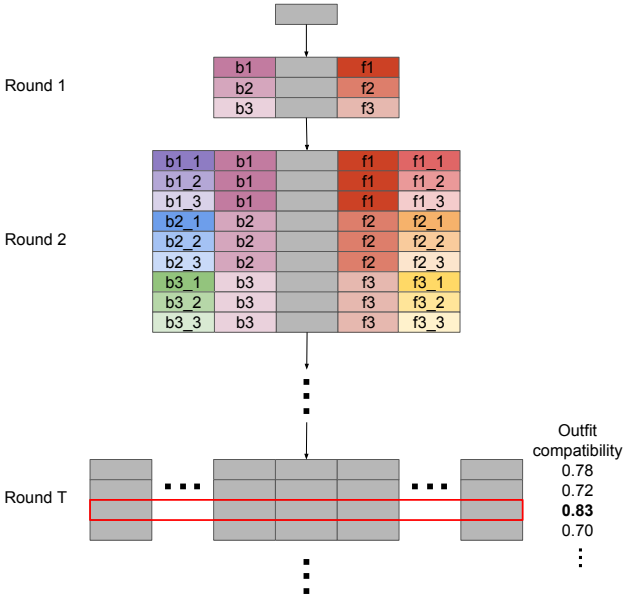
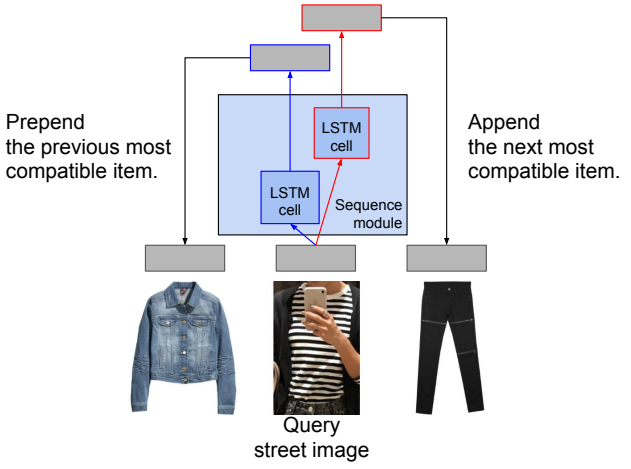
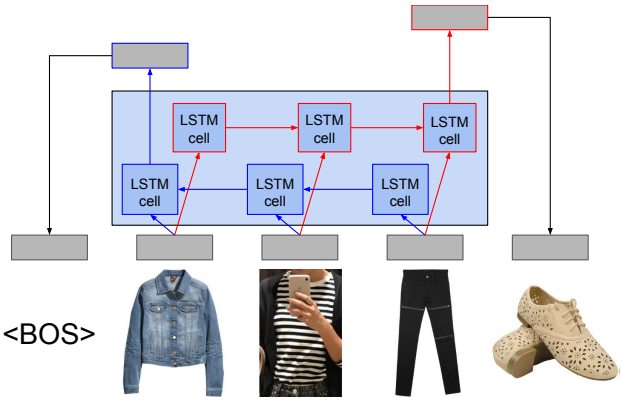


Fig. 10 Example of the 3-width beam search. Each box is an image feature. The gray box at the top is that of a street image. The features with **f** are added forward, while those with **b** are added backward. The number in a box means the rank of the garment, *e.g.*, **f2** means the second most compatible garment added forward in the first round, and **b1_3** means the third most compatible garment next to **b1** in the second round. Each row in a round is an outfit candidate to recommend. After the final round T, the most compatible outfit (surrounded by a red frame) is selected from all candidates by their compatibility.



(a) Round 1.



(b) Round 2.

Fig. 9 Example of the outfit recommendation S2S. In each round of the task, the recommendation models receive a street image (only in the first round) or the intermediate outfit (the street image and added shop images), and then add a compatible shop image to each side of the outfit. They repeat such rounds until they add the *BOS* (beginning of a sequence) and the *EOS* (end of a sequence).

Fig. 11 Recommended outfits using Polyvore shop images by Bi-LSTM-S2S and Bi-LSTM, given a WEAR street image input.

5. CONCLUSION

The previous image-based outfit recommendation models have a problem handling street image inputs. Thus we proposed Bi-LSTM-S2S to solve the problem, reducing the gap between street and shop image features by the metric learning. We confirmed the effectiveness of our model with four

tasks including two newly introduced tasks: FITB-S2S and outfit recommendation S2S. We concluded that Bi-LSTM-S2S can more appropriately handle street image inputs than Bi-LSTM. It is also expected that the S2S module can be applied to other image-based outfit recommendation models.

6. ACKNOWLEDGEMENTS

I would like to greatly thank T. Nakamura for insightful discussions, feedback, and teaching how to research and write technical documents. In discussions with you, you often told me "It's a nice idea.". You encouraged me in my work. I owe what I am to you. I got to like research thanks to you.

I would like to thank R. Goto for feedback. You told me "Ideas are often denied, but you should continue to think and bring ideas." You inspired me to continue my research.

I would like to thank T. Yamanaka for feedback, computing resources, permission for the change of my lab and theme.

I would like to thank ZOZO, Inc. for the data from ZOZOTOWN and WEAR.

I would like to thank all users of WEAR for posting snap images.

References

[1] 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 *Proceedings of the IEEE international conference on computer vision*, pages 3343–3351, 2015.

[2] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. Learning fashion compatibility with bidirectional lstms. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1078–1086. ACM, 2017.

[3] Ruining He, Charles Packer, and Julian McAuley. Monomer: Non-metric mixtures-of-embeddings for learning visual compatibility across categories. *CoRR*, 2016.

[4] Junshi Huang, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *Proceedings of the IEEE international conference on computer vision*, pages 1062–1070, 2015.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Julia Lasserre, Katharina Rasch, and Roland Vollgraf. Studio2shop: from studio photo shoots to fashion articles. *arXiv preprint arXiv:1807.00556*, 2018.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[8] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016.

[9] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

[10] Takuma Nakamura and Ryosuke Goto. Outfit generation and style extraction via bidirectional lstm and autoencoder. *arXiv preprint arXiv:1807.03133*, 2018.

[11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[12] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[13] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4642–4650, 2015.

[14] Shuai Zheng, Fan Yang, M Hadi Kiapour, and Robinson Piramuthu. Modanet: A large-scale street fashion dataset with polygon annotations. *arXiv preprint arXiv:1807.01394*, 2018.