



Garment Recommendation with Memory Augmented Neural Networks

Lavinia De Divitiis , Federico Becattini , Claudio Baecchi ,
and Alberto Del Bimbo  

University of Florence, Florence, Italy
{lavinia.dedivitiis, federico.becattini,
claudio.baecchi, alberto.delbimbo}@unifi.it

Abstract. Fashion plays a pivotal role in society. Combining garments appropriately is essential for people to communicate their personality and style. Also different events require outfits to be thoroughly chosen to comply with underlying social clothing rules. Therefore, combining garments appropriately might not be trivial. The fashion industry has turned this into a massive source of income, relying on complex recommendation systems to retrieve and suggest appropriate clothing items for customers. To perform better recommendations, personalized suggestions can be performed, taking into account user preferences or purchase histories. In this paper, we propose a garment recommendation system to pair different clothing items, namely tops and bottoms, exploiting a Memory Augmented Neural Network (MANN). By training a memory writing controller, we are able to store a non-redundant subset of samples, which is then used to retrieve a ranked list of suitable bottoms to complement a given top. In particular, we aim at retrieving a variety of modalities in which a certain garment can be combined. To refine our recommendations, we then include user preferences via Matrix Factorization. We experiment on IQON3000, a dataset collected from an online fashion community, reporting state of the art results.

Keywords: Garment recommendation · Memory Augmented Neural Networks · Recommendation systems

1 Introduction

Recommendation systems are tools that have recently found pervasive use across several application fields. Ranging from social networks to online stores and streaming services, whenever a user has the ability to sift through large databases, machine learning algorithms can come to the aid by suggesting items of possible interest. A field that is heavily hinged on the effectiveness of recommendation systems is the fashion industry. In fact, revenues for large companies nowadays mostly come from online stores, where users are free to search for desired items. To maximize incomes, it is important for the seller to be able to suggest relevant items, often also exploiting advertisements specifically targeted on user profiles.

© Springer Nature Switzerland AG 2021

A. Del Bimbo et al. (Eds.): ICPR 2020 Workshops, LNCS 12662, pp. 282–295, 2021.

https://doi.org/10.1007/978-3-030-68790-8_23

Two typical strategies for recommendation are content based and collaborative filtering. The former is based on suggesting similar content to what the user has already purchased in the past. The latter instead exploits a pool of user preferences to infer what a certain individual might like, based on what similar users have purchased. Both these approaches have limitations. In particular, they can easily end up suggesting several variations of similar items due to their overall popularity or simply because they are close to items owned by the user [13]. This might even be counterproductive, since it limits the visibility of other items which are relevant yet novel or unexpected.

In this paper we propose a recommendation system capable of generating a variety of fashion items compatible with a given garment. In particular, we focus on generating different modalities to compose an outfit, rather than suggesting redundant and similar items. To achieve this, we exploit a persistent Memory Augmented Neural Network (MANN), which has proven effective to model diversity [17, 18]. This kind of models finds its strength in the usage of an external memory, where samples can be explicitly stored at training time and then be read at inference time. Thanks to a trainable memory controller, we are able to store representations of non-redundant outfits, which are then retrieved at inference time to provide recommendations.

For outfit here we refer to a top-bottom pair, e.g. a t-shirt and a skirt. The problem we address is in fact the one of suggesting suitable bottom items, given a top as input. However, differently from prior work [26], we cast the problem as a retrieval task, rather than simply assessing the compatibility between garments.

The main contributions of this paper are the following:

- We propose a garment recommendation system based on Memory Augmented Neural Networks. To the best of our knowledge, we are the first to adopt MANNs for garment recommendation.
- We train our model to propose a variety of bottoms instead of a set of redundant and almost identical garments, shifting the focus of the retrieval task from single items to garment matching modalities.
- We obtain state of the art results on the IQON3000 dataset [26], a collection of outfits collected from an online fashion community.

2 Related Work

2.1 Recommendation Systems

In recommendation systems, collaborative filtering methods are often employed to find the relationship between two sets of entities, such as users and products. Among collaborative filtering methods, Matrix Factorization (MF) determines latent features among these sets to find similarities and allows to make predictions on both entities. In [15] they use a variant of Matrix Factorization called Non-negative Matrix Factorization (NMF), for decomposing multivariate data and study two different algorithms. In [12] the authors employ MF for Netflix movie recommendation to users. They incorporate additional information such

as implicit feedback, temporal effects and confidence levels. On a more general scale, [10] perform an introductory study on collaborative filtering and MF for product recommendation. They also propose a way to measure the effectiveness of recommender systems. In a similar fashion, [22] illustrate various recommender systems putting particular attention into the difference of the various approaches. Focusing their research on Non-negative MF, [28] perform an extensive survey on different methods using NMF, analysing the design principles, characteristics, problems, relationships, and evolution of the algorithms. Finally [2] perform an evolutionary survey on MF and collaborative filtering focusing on the future implications of these methods.

Many recommender systems employ MF for recommending garment matches. In [3] they propose a multi-view Non-negative Matrix Factorization (NMF) to solve the problem of clothing matching leveraging multiple types of features. [25] propose a content-based neural scheme to model the compatibility between fashion items to give meaningful suggestions to the user. In a scenario where multiple items have to be recommended, [30] propose several extensions of MF to predict user ratings on garment packaging composed of tops and bottoms. Moreover, they also create a package recommendation dataset to serve their task. Finally, [6] propose a way to interpret MF based clothing matching proposals, giving also suggestions on how to interpret them and on how to modify those negative garments to make them appealing to the user. They employ Non-negative MF to suggest the alternatives for each fashion item pair.

2.2 Garment Recommendation

In the world of recommendation systems, Garment Recommendation is the task of generating meaningful recommendations of garments based on user interests or on a given input garment. [8] experiment on the *Amazon clothing dataset* to recommend clothes to people based in their interests by using Matrix Factorization techniques. Similarly, in [9] they give the users suggestions on clothes but considering a set of clothes instead of a single piece. Differently to [8], they use a functional tensor factorization method to model the interactions between user and fashion items. [24] leverage the real-world FashionVC dataset to perform complementary clothing matching by integrating deep neural networks and the rich fashion domain knowledge. Recently [1] leveraged human pose and behaviour to give meaningful suggestions to the user when no prior information about the user is known.

Remaining in the garment recommendation domain, another task that has recently gained interest: Virtual try-on. Virtual try-on complements the recommendation systems with a way of showing the end user how one or more clothes would look if “tried on”, simulating the real life act of trying a garment before buying. Here [7] leverage 3D cameras to recreate a 3D model of the user and apply on it the selected garment. In [11] they investigate online apparel shoppers’ use of Virtual Try-on to reduce product risks and increase enjoyment in online shopping. The research is based on the electronic Technology Acceptance Model (e-TAM). Finally [32] introduce a new try-on setting, which enables to

change both the clothing item and the person’s pose. They use a pose-guided virtual try-on scheme based on a generative adversarial network (GAN).

2.3 Memory Augmented Networks

Memory Augmented Neural Networks (MANN) are a particular declination of Neural Networks that exploit a controller network with an external memory, in which samples can be explicitly stored. These models have been originally introduced to solve algorithmic tasks [4, 23, 27, 29], however several applications of MANNs have been proposed in literature [16, 18–20, 31]. The first work to propose a model equipped with an external memory has been Neural Turing Machines (NTM) [4]. The authors exploited the MANN as a working memory to solve data manipulation tasks, such as sorting or copying.

The problem of online learning has also been addressed with MANNs. Pernici *et al.* [20] rely on an external memory to incrementally store people identities for re-identification purposes. In [31], Memory Augmented Networks have been shown to aid object tracking, by offering the ability to store templates of the tracked objects. Also incremental classifiers have been implemented [21], by adding classes in memory, incrementally.

Several works have found MANNs beneficial for Question Answering [27, 29] or Visual Question Answering [14, 16] tasks. In these works, some attention mechanism is usually exploited to guide the answering process by sequentially attending to different memory locations or to assign a different importance to common and uncommon question answer pairs.

Most of these approaches, however, rely on episodic memories. This means that the model learns how to manipulate data, rather than learning to build a persistent collection of samples. Recently, a model for trajectory forecasting has been proposed by Marchetti *et al.* [17, 18], where the model learns to store non redundant samples relying on a specifically tailored writing controller. In this work, we rely on a similar strategy, by adding in memory a collection of garments. We then use these to retrieve different modalities to create an outfit by matching clothing items. Differently from [17], we make use of a convolutional model to store image embeddings in memory and we cast the model to solve a recommendation task by adding user information via Matrix Factorization.

3 Garment Recommendation with MANNs

We focus on the problem of garment recommendation, i.e. the task of retrieving a ranked list of bottoms that can be combined with a given top. Tops and bottoms are clothing items that can be worn together, defining an outfit. We refer to the set of top and bottom images respectively as $\mathcal{T} = \{t_0, t_1, \dots, t_{N_T}\}$ and $\mathcal{B} = \{b_0, b_1, \dots, b_{N_B}\}$, where N_T and N_B are the total number of garments in the two sets. Each user $u_k \in \mathcal{U} = \{u_0, u_1, \dots, u_{N_U}\}$ is associated with a collection of outfits, i.e. top-bottom pairs which have been rated as positive matches.

We indicate this outfit set for user u_k as $\mathcal{O}_k = \{(t_i, t_j)\}_{ij}$, where i and j act as indexes selecting items in \mathcal{T} and \mathcal{B} .

The goal of the garment recommendation task can therefore be formulated as the task of retrieving the bottom t_j given a user u_k and a top t_i belonging to \mathcal{O}_k .

In the following, we first present our Memory Augmented Neural Network based method for establishing matches according to general style preferences and we then outline our recommendation refinement strategy, which includes personal user preferences via Matrix Factorization.

3.1 Feature Representation

In order to let our model manipulate tops and bottoms and store them into its external memory, we first need to devise a way to extract compact representations. To this end, we train an autoencoder that jointly learns to reconstruct top and bottom items forming an outfit.

Separate convolutional encoders project top and bottom images into a common latent space. We denote the top encoding as τ and the bottom encoding as β . The two features are then blended together with a scalar product into a joint representation ϕ and are then separately reconstructed with two deconvolutional decoders. Figure 1 depicts the structure of the model. In our experiments we observed that training the top and bottom autoencoders jointly led to a better generalization, reducing overfitting.

The benefits of using an autoencoder are twofold: on the one hand, we are able to obtain a compact representation of garments without any further annotation cost; on the other hand, we are also training a decoding function which will be useful for extracting bottoms from memory.

We use the same architecture for the two pairs of encoders and decoders. As for the encoder, we perform three 3×3 2D convolutions with 8, 16 and 32 filters respectively. All three convolutions have a padding applied of 1 pixel. We then apply 3×3 max pooling with a stride of 3. The relu activation function is applied after each convolution. On the decoder side, we employ three 2D transposed convolutions to attempt to reconstruct the input. Specifically, we set the number of channels to 16, 8 and 3 with a kernel size of 5×5 , 8×8 and 3×3 respectively. As for padding we apply 4, 2 and 3 pixels respectively, with an extra pixel of output padding in the first transposed 2D convolution. The relu activation function is applied after each transposed convolution. Finally to each encoder-decoder pair output we apply the sigmoid function to output values in the range $[0, 1]$.

3.2 Model

To find suitable bottom garments to be paired with a given top, we adopt a model based on a Memory Augmented Neural Network. The model is trained to identify a non-redundant subset of outfits to be stored in memory, which

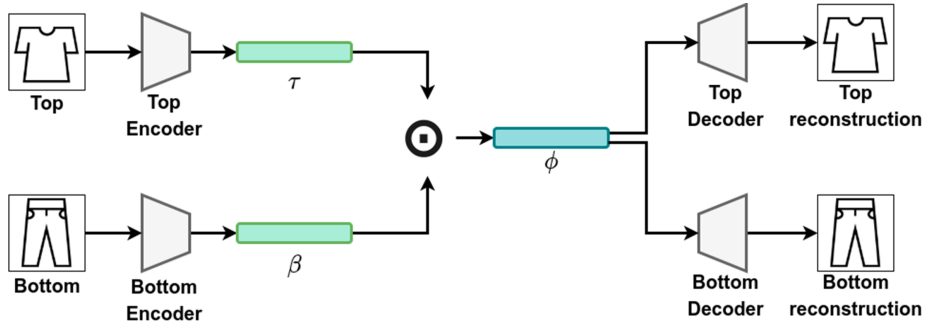


Fig. 1. Top and bottom images are separately encoded using convolutional encoders. The resulting features τ and β are then combined through a scalar product into a common representation ϕ . Finally, ϕ is decoded into two separate reconstructions with deconvolutional layers to obtain an approximation of the inputs.

can then be used to produce recommendations. The memory acts as a learnable interface between user queries and the underlying database. Since our aim is to suggest multiple outfit modalities, rather than retrieving all suitable instances in the dataset, we train the model to store only samples that can be used to suggest a variety of bottoms.

Given an outfit composed of a top-bottom pair (t, b) , we want our model to output a ranked list of bottoms, where at least one is sufficiently close to b , when t is fed as input. To this end, when the top t is fed to the model, we encode it into a feature representation τ , using the encoding from Sect. 3.1. This compact representation of the input is then used as key to access the external memory. Memory access is performed via cosine similarity between the key and the top component of each stored outfit. This generates a distribution of similarity scores s_i for each element in memory:

$$s_i = \frac{\tau \cdot \tau_i}{\|\tau\| \cdot \|\tau_i\|} \quad \text{for } i = 0, \dots, |M| - 1 \quad (1)$$

where $|M|$ is the memory size.

To generate a ranked set of recommendations, we take the K_N samples that give the highest scores. For each outfit retrieved from memory, we take its bottom β_{K_i} and we combine it with τ via dot product, to obtain a joint representation ϕ_{K_i} . Each feature is then fed to the decoder part of the autoencoder (see Sect. 3.1) to reconstruct different bottom images. Each reconstruction is performed in parallel, independently from the others. Figure 2 shows the inference process of our model.

To guarantee diversity, as well as satisfactory reconstructions, we train a writing controller that decides what samples to insert into memory. At training time, a top is fed to the network and K_N bottoms are proposed as output, as described above. The task of garment recommendation is inherently multimodal, i.e. multiple outputs might be considered correct given a single input. In other

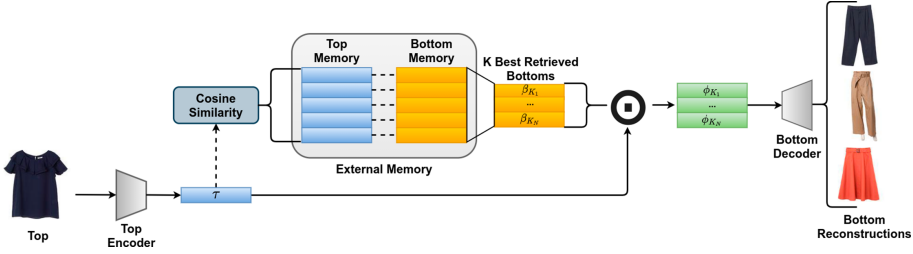


Fig. 2. Given an input image depicting a top, we encode it into a latent representation τ and use it to retrieve samples from memory via cosine similarity. The bottom part of the best samples are then combined with τ and decoded into bottom reconstructions.

terms, a user might consider multiple bottoms to be a good match with a top to create different outfits.

Since we only have one ground truth bottom t^* , we cannot explicitly add supervision on all the generated outputs. To overcome this limitation, we borrow a concept from the trajectory forecasting literature, where the multimodality of the task plays an even more central role. In fact, we derive our controller loss from the Variety Loss [5] to train our network. The Variety Loss, often referred to as Best-of-K loss, encourages the generation of diverse samples \hat{t}_k by simply backpropagating the error only for the best prediction, leaving the other predictions free to explore the output space:

$$\mathcal{L}_{Variety} = \min_k \|\hat{t}_k - t^*\|_2 \tag{2}$$

Our memory writing controller, i.e. a trainable feedforward network that outputs a writing probability P_w for a given sample, takes as input the error value given by the Variety Loss itself. Based on the best reconstruction error, in fact, it establishes whether the current sample should be written to memory. The rationale behind this idea is that a sample only needs to be written if the model is not capable of generating a sufficiently close reconstruction of the ground truth.

We therefore optimize the controller loss $\mathcal{L}_{Controller}$ defined as follows:

$$\mathcal{L}_{Controller} = \mathcal{L}_{Variety}(1 - P_w) + (1 - \mathcal{L}_{Variety})P_w \tag{3}$$

A similar loss has been adopted in [17] to fill a persistent memory with non-redundant samples. The controller generates a writing probability in $[0, 1]$ and, since we use normalized images and an MSE error function for the Variety Loss, $\mathcal{L}_{Variety}$ also yields values in $[0, 1]$. When the reconstruction generates a high error, then $\mathcal{L}_{Variety}$ will be close to 1 and the sample should be written to improve the recommendation capabilities of the network. In this case the loss will optimize the controller to output a high writing probability. On the contrary, when the reconstruction error is low, the controller will be optimized to discard the observed outfit.

When a sample gets written in memory, we store the pair of encodings for the top and bottom. At training time, we start from a randomly initialized memory with K_N samples, to be able to always suggest a sufficient number of elements. We empty the memory after each epoch and populate the final memory after the controller has been trained by iterating once again on the training set.

3.3 General Recommendations and User Preferences

The model presented so far is capable of retrieving bottoms that can be matched with a given top. These matches, however, follow the general fashion criteria underlying the data distribution. Effective recommendation systems typically strive at suggesting items that are targeted for specific users, based on some past information, such as purchase history, or similarity with other users.

A common approach to generate targeted recommendations is Matrix Factorization [2, 28]. In our garment recommendation system, we first retrieve a set of bottoms with our Memory Augmented Neural Network. These bottoms represent different modalities to compose an outfit based on the given top. Although the MANN is capable of providing an ordered set according to top similarity with stored samples, this ranking does not take into account the user.

We refine our recommendations by re-ranking the set of bottoms via Matrix Factorization, which generates compatibility scores between the user and each bottom. Matrix Factorization projects the user and bottom IDs into a common space using trainable encoding functions and combines them together with a dot product. For both the user and the proposed bottoms, we also learn a bias to shift the decision. The final compatibility score, with values in $[0, 1]$, is obtained through a sigmoid activation.

To be able to work with known garments, i.e. belonging to a given database collection, we retrieve the most similar bottoms in the training set to the bottoms reconstructed by the decoder of the MANN. This allows us to learn a meaningful projection for bottoms, based on all user preferences.

4 Experiments

In the following we first provide some details on the dataset and metrics used to evaluate our model and then we report results, comparing the proposed approach with the state of the art.

4.1 Dataset and Metrics

For our experiments, we use the IQON3000 dataset [26]. This dataset contains a collection of 308,747 outfits, handpicked by 3,568 different users. Each outfit comprises garments belonging to 6 categories: *coat*, *top*, *bottom*, *one piece*, *shoes* and *accessories*. In total the dataset has 672,335 fashion items. All items are labeled with attributes such as *color*, *category*, *price* and *description*. Similarly to prior work [26], in this paper we focus solely on top and bottom garments and

Table 1. Comparison between our model and GP-BPR [26]. An item is considered correct if it shares Category and/or Color with the ground truth.

Num items	Category \times Color		Category		Color	
	Ours	GP-BPR	Ours	GP-BPR	Ours	GP-BPR
5	30	27	81	75	58	55
10	45	44	89	84	73	70
20	59	56	93	90	85	81
30	67	63	95	92	91	87
40	71	69	96	93	94	90
50	75	72	96	94	96	92
60	78	74	97	94	97	92

we always consider tops to be given as inputs in order to find matching bottoms. The proposed model, however, could be easily adapted to the inverse problem by simply swapping tops and bottoms, without any further modification.

The dataset was first introduced to address the task of personalized garment matching. This was originally engineered to simply assess whether a model was capable of assigning a higher score to known positive matches compared to random ones. Here we extend this task, aiming at obtaining a ranked list of recommendations among all possible garments. However, we believe that an effective recommendation system should propose a variety of different outputs instead of suggesting several small variations of the same garment.

To this end, instead of simply comparing the ID of the retrieved bottoms, we compare their categories and colors. Proposing bottoms of different colors and categories, to be paired with the same top, means that we are proposing diverse modalities to combine the given top and generate an outfit. In the dataset there are 62 bottom categories, 12 different colors and a total of 742 combinations of the two.

We evaluate our approach by measuring the Accuracy@K, i.e. the fraction of samples for which the model suggests at least one bottom sharing the same category and/or color of the ground truth, among the first K retrieved items. In addition, we report the quality of the ranking by measuring the mean Average Precision (mAP). This is of interest especially when considering user personal preferences.

4.2 Results

Here we discuss the results obtained by our model. First of all, we evaluate the accuracy of the model by considering colors and categories, as explained in Sect. 4.1. We measure the accuracy varying the number of retrieved items, as shown in Table 1. Despite the number of categories is approximately five times higher than the number of colors, the system immediately obtains an

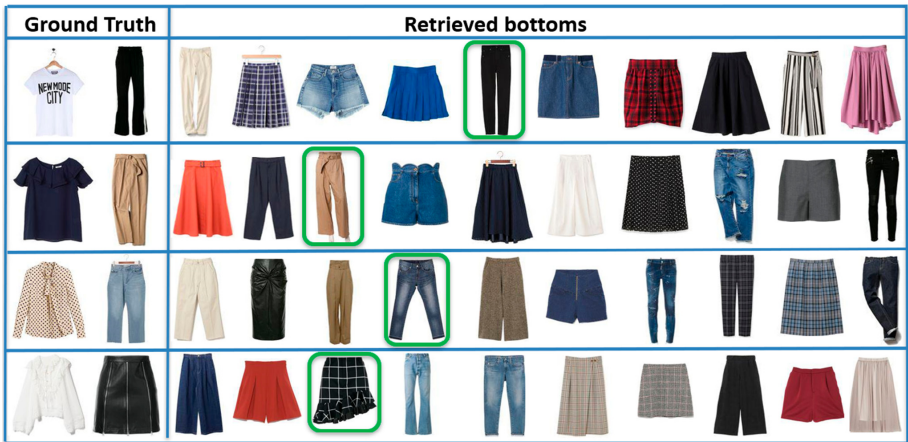


Fig. 3. Qualitative results generated by our proposed model. The top is given as input and a ranked set of bottom is retrieved. It can be observed that bottoms similar to the ground truth appear among the first recommendations.

accuracy above 80%, even with just 5 retrieved items. Considering colors, instead, the accuracy starts from around 60%. We have attributed this to the fact that color matching, despite following precise social rules, allows some flexibility with similar tonalities.

As expected, when considering both color and category, the accuracy drops considerably. Interestingly, in all cases, we are able to obtain a higher accuracy than the state of the art. In particular, we report results obtained by GP-BPR [26]. The results are obtained with the pre-trained models released by the authors by generating all compatibility scores between the pair of current top and user and all the bottoms in the dataset. We then take the highest ranking garments to generate the recommendations.

Despite GP-BPR explicitly considers the compatibility with each bottom in the dataset, our approach is still able to yield a higher accuracy relying only on the subset of garments stored in memory. Figure 3 shows some qualitative samples of bottoms retrieved from memory. It can be seen how there is a high variability in the suggested outfits and that a similar bottom to the ground truth is found in the first positions of the rankings.

We then measure the mean Average Precision obtained by our system. Figure 4 shows the curve, varying the number of retrieved garments. The combination of the Memory Network with Matrix Factorization yields to better results than GP-BPR [26] for a low number of recommended items, which is the scenario with the most relevant applicability.

Finally we show in Fig. 5 and Fig. 6 the effect of removing user information in the recommendation pipeline. In this way, we obtain only recommendations that follow a common sense, rather than personalized suggestions. Again, we compare our results against [26], removing the user information also for this model for

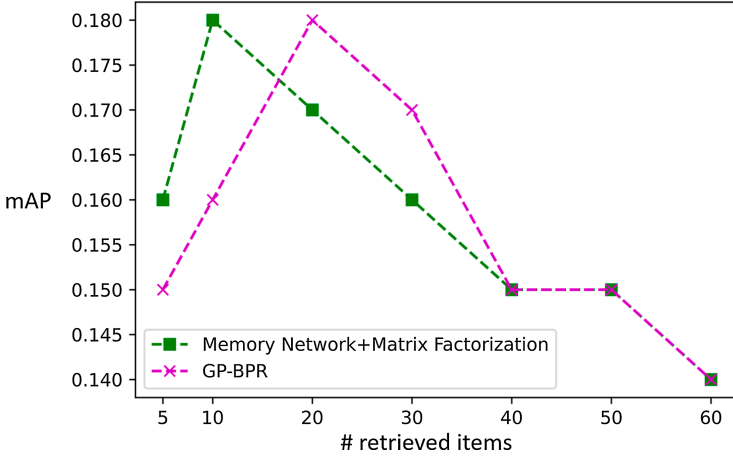


Fig. 4. Mean Average Precision of the proposed approach using Memory Network and Matrix Factorization. As reference, we report the results obtained by GP-BPR [26]. Interestingly, with a low number of retrieved items, our method obtained a higher mAP compared to GP-BPR.

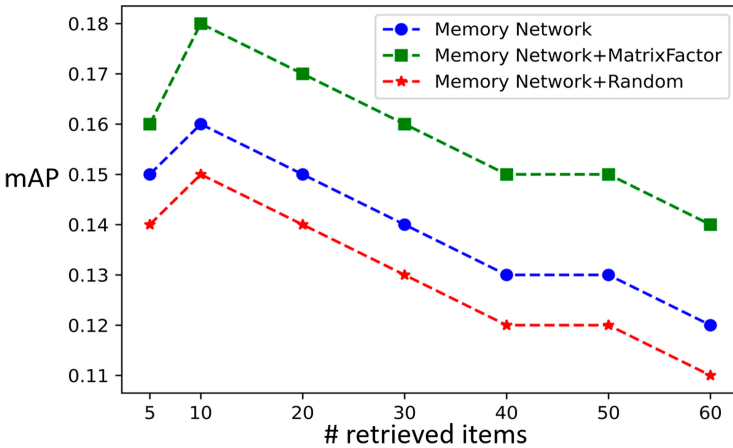


Fig. 5. Impact of reranking on the garments retrieved from the Memory Network. Matrix Factorization provides an improvement including user preferences.

a fair comparison. It can be seen in Fig. 5 how our method reports a 2% drop, while on the other hand, GP-BPR suffers from a much higher loss (Fig. 6). This suggests that our memory network is capable of providing a satisfactory set of garments even without considering the user. At the same time, we verify that Matrix Factorization is capable of reranking the garments effectively and that overall the Memory Network, with and without Matrix Factorization, are capable or providing better results compared to a Random reranking baseline.

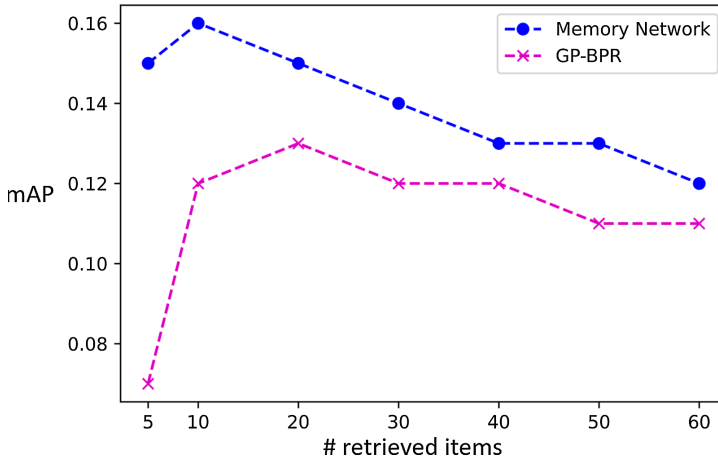


Fig. 6. Comparison of our proposed method against GP-BPR [26] without including user information.

5 Conclusions

In this paper we have presented a garment recommendation system based on a Memory Augmented Neural Network. The model relies on an external memory, which is filled by a trainable controller to store non redundant samples that capture multiple modalities to match top and bottom garments. In this way, the system is capable of retrieving and suggesting a variety of garments instead of proposing small variations of the same outfit. We have shown how this reflects on improved recommendation capabilities compared to the state of the art.

Acknowledgement. This work was partially supported by the Italian MIUR within PRIN 2017, Project Grant 20172BH297: I-MALL - improving the customer experience in stores by intelligent computer vision.

References

1. Bigi, W., Bacchi, C., Del Bimbo, A.: Automatic interest recognition from posture and behaviour. In: Proceedings of the 28th ACM International Conference on Multimedia, MM 2020, pp. 2472–2480. Association for Computing Machinery, New York (2020). <https://doi.org/10.1145/3394171.3413530>
2. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. *Knowl.-Based Syst.* **46**, 109–132 (2013)
3. Chang, W.Y., Wei, C.P., Wang, Y.C.F.: Multi-view nonnegative matrix factorization for clothing image characterization. In: 2014 22nd International Conference on Pattern Recognition, pp. 1272–1277. IEEE (2014)
4. Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. arXiv preprint [arXiv:1410.5401](https://arxiv.org/abs/1410.5401) (2014)

5. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2255–2264 (2018)
6. Han, X., Song, X., Yin, J., Wang, Y., Nie, L.: Prototype-guided attribute-wise interpretable scheme for clothing matching. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 785–794 (2019)
7. Hauswiesner, S., Straka, M., Reitmayr, G.: Free viewpoint virtual try-on with commodity depth cameras. In: Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry, pp. 23–30 (2011)
8. He, R., McAuley, J.: VBPR: visual Bayesian personalized ranking from implicit feedback. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
9. Hu, Y., Yi, X., Davis, L.S.: Collaborative fashion recommendation: a functional tensor factorization approach. In: Proceedings of the 23rd ACM International Conference on Multimedia, pp. 129–138 (2015)
10. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems: An Introduction. Cambridge University Press, Cambridge (2010)
11. Kim, J., Forsythe, S.: Adoption of virtual try-on technology for online apparel shopping. *J. Interact. Mark.* **22**(2), 45–59 (2008)
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
13. Kotkov, D., Veijalainen, J., Wang, S.: How does serendipity affect diversity in recommender systems? A serendipity-oriented greedy algorithm. *Computing* **102**(2), 393–411 (2020)
14. Kumar, A., et al.: Ask me anything: dynamic memory networks for natural language processing. In: International Conference on Machine Learning, pp. 1378–1387 (2016)
15. Lee, D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems 13, pp. 556–562 (2000)
16. Ma, C., et al.: Visual question answering with memory-augmented networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6975–6984 (2018)
17. Marchetti, F., Becattini, F., Seidenari, L., Bimbo, A.D.: Mantra: memory augmented networks for multiple trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7143–7152 (2020)
18. Marchetti, F., Becattini, F., Seidenari, L., Del Bimbo, A.: Multiple trajectory prediction of moving agents with memory augmented networks. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020)
19. Pernici, F., Bartoli, F., Bruni, M., Del Bimbo, A.: Memory based online learning of deep representations from video streams. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2324–2334 (2018)
20. Pernici, F., Bruni, M., Del Bimbo, A.: Self-supervised on-line cumulative learning from video streams. *Comput. Vis. Image Underst.* 102983 (2020)
21. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: ICARL: incremental classifier and representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2001–2010 (2017)

22. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) *Recommender Systems Handbook*, pp. 1–35. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_1
23. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: *International Conference on Machine Learning*, pp. 1842–1850 (2016)
24. Song, X., Feng, F., Han, X., Yang, X., Liu, W., Nie, L.: Neural compatibility modeling with attentive knowledge distillation. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 5–14 (2018)
25. Song, X., Feng, F., Liu, J., Li, Z., Nie, L., Ma, J.: Neurostylist: neural compatibility modeling for clothing matching. In: *Proceedings of the 25th ACM International Conference on Multimedia*, pp. 753–761 (2017)
26. Song, X., Han, X., Li, Y., Chen, J., Xu, X.S., Nie, L.: GP-BPR: personalized compatibility modeling for clothing matching. In: *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 320–328 (2019)
27. Sukhbaatar, S., Weston, J., Fergus, R., et al.: End-to-end memory networks. In: *Advances in Neural Information Processing Systems*, pp. 2440–2448 (2015)
28. Wang, Y.X., Zhang, Y.J.: Nonnegative matrix factorization: a comprehensive review. *IEEE Trans. Knowl. Data Eng.* **25**(6), 1336–1353 (2012)
29. Weston, J., Chopra, S., Bordes, A.: Memory networks. *arXiv preprint [arXiv:1410.3916](https://arxiv.org/abs/1410.3916)* (2014)
30. Wibowo, A.T., Siddharthan, A., Lin, C., Masthoff, J.: Matrix factorization for package recommendations. In: *Proceedings of the RecSys 2017 Workshop on Recommendation in Complex Scenarios (ComplexRec 2017)*. CEUR-WS (2017)
31. Yang, T., Chan, A.B.: Learning dynamic memory networks for object tracking. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 152–167 (2018)
32. Zheng, N., Song, X., Chen, Z., Hu, L., Cao, D., Nie, L.: Virtually trying on new clothing with arbitrary poses. In: *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 266–274 (2019)